VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Analýza vlastností speciálních bloků z knihovny programu Matlab-Simulink pro řízení elektrických pohonů

Property Analysis of Special Blocks from MATLAB-Simulink Library for Control of Electric Drives

2013 Patrik Kováč

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání bakalářské práce

Student: **Patrik Kováč**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2602R014 Aplikovaná a komerční elektronika

Téma: Analýza vlastností speciálních bloků z knihovny programu Matlab-Simulink pro řízení elektrických pohonů
Property Analysis of Special Blocks from MATLAB-Simulink Library for Control of Electric Drives

Zásady pro vypracování:

1. Rozeberte možnosti programu Matlab-Simulink pro simulaci řízení elektrických pohonů
2. Popište použitelné standardní bloky z knihovny Simulinku sloužící pro simulaci řízení elektrických pohonů
3. S použitím výše uvedených standardních bloků proveďte simulaci zadaných úloh řízení elektrických pohonů
4. Sestavte vlastní model regulátoru, proveďte simulace a výsledky srovnejte s bodem 3

Seznam doporučené odborné literatury:

Dle pokynů vedoucího závěrečné práce

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Ivo Neborák, CSc.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

ČESTNÉ PROHLÁŠENÍ

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal."

7.května 2013

.....

Podpis

PODĚKOVÁNÍ

Na tomto místě bych velice rád poděkoval vedoucímu bakalářské práce doc. Ing. Ivu Neborákovi, CSc. za odborné konzultace a rady, které mi pomohly při vypracování této bakalářské práce. Rád bych poděkoval i své manželce za podporu při studiu.

ABSTRAKT

Tato bakalářská práce se zabývá analýzou vlastností speciálních bloků z knihovny programu Matlab-Simulink pro simulaci řízení elektronických pohonů. Práce obsahuje teoretickou část, kde jsou popsány vybrané bloky pro řízení elektronických pohonů z uvedeného programu pro spojité a diskrétní systémy.
V další části jsou vlastní simulace s popsaných bloků, se zaměřením na omezení integrační složky pomocí standardních bloků a také pomocí hotového bloku PI regulátoru a metody back – calculation. Poslední simulace se zabývá simulací přechodové charakteristiky v závislosti na změně filtračního koeficientu derivační složky.

Klíčová slova

Matlab - Simulink, omezení integrační složky, PI regulátor, back-calculation, simulace

ABSTRACT

This Bachelor´s thesis deals with analysis the properties of special blocks from library of Matlab-Simulink for control of electric drives. This work contains the theoretical part where are described choice blocks for control of electric drives from the program for continuous and discrete systems.
In the next part of my work are own simulations from choice blocks with a focus on limitations of the integrator using standard blocks and complete block PI controller too and using method back – calculation. The last simulation deals simulation transient characteristic in response to changes in filtration coefficient of derivative component.

Key words

Matlab - Simulink, limitations of the integrator, PI controller, back-calculation, simulation

Seznam použitých symbolů a zkratek:

e – regulační odchylka

e_s – odchylkový signál

F_R – přenos soustavy

K_P – zesílení proporcionálního členu

u – akční veličina

u_r – výstup regulátoru

u_{lim} - saturační hodnota řízení

w – řídící veličina

y – regulovaná veličina

Obsah

1. Úvod

Elektrické pohony jsou vědním oborem, který se zabývá využitím elektrických strojů pro elektromechanickou přeměnu energie a řízením této přeměny. Úkolem elektrického pohonu jako průmyslového zařízení je uvést poháněný pracovní mechanismus předepsaným způsobem do určitého pohybového stavu, určeného jeho funkcí tak, aby byla pracovním mechanismem provedena požadovaná technologická operace nebo zajištěn určitý technologický proces.

Řízením pohonů rozumíme jejich ovládání a regulaci. Ovládání je takové řízení, při němž jsou v určitém předepsaném sledu prováděny logické operace s případnou indikací stavu pohonu na základě popudu obsluhy nebo nadřazeného řídícího systému.Důležité přitom je, že pohon pracuje bez jakékoliv zpětné vazby. Regulace je pak takové řízení, při němž pohon díky zpětné vazbě pracuje na takových mechanických charakteristikách, aby regulovaná veličina sledovala s požadovanou přesností a dynamikou řídící veličinu. [4]

V této bakalářské práci jsem se prvně zaměřil na popis základní bloků z programu Matlab – Simulink pro řízení elektrických pohonů ve spojité a diskrétní formě a z těchto bloků jsem posléze sestavil modely, které představují proporcionálně-integrační regulátor (PI regulátor), který patří k nejpoužívanějším typům regulátorů v technice elektrických pohonů . Dále jsem se v práci zaměřil na simulaci PI regulátoru s omezením integrační složky a pomocí několika metod jsem porovnal výstupní grafy, abych zjistil chování omezení jednak regulátoru složeného z popsaných základních bloků a také už vytvořených hotových bloků z programu Matlab – Simulink. V poslední části simulace jsem si vyzkoušel chování PID regulátoru v závislosti na změně hodnoty filtračního koeficientu - N, který je vázán na derivační složku regulátoru. Nasimuloval jsem dvě přechodové charakteristiky pro dvě různé hodnoty N a zkoumal chování v závislosti právě na filtračním koeficientu.

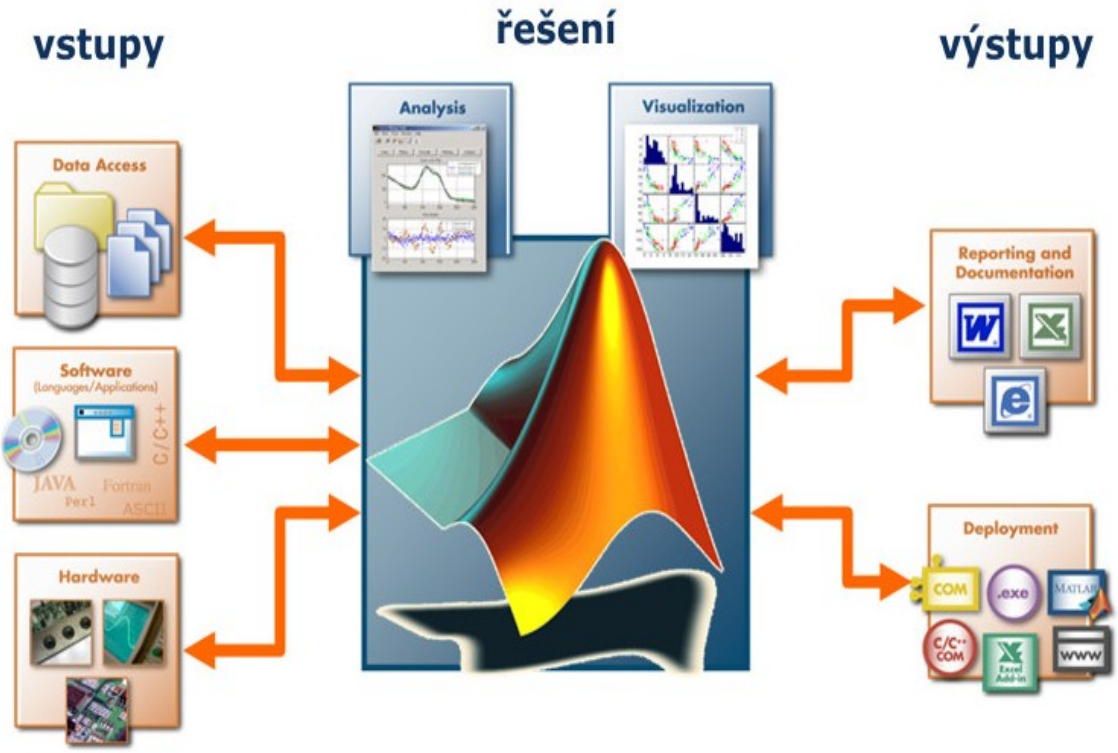
2. Popis programu Matlab-Simulink

2.1 Matlab

Matlab je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. Matlab je nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací.

Matlab poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti.

Otevřená architektura Matlabu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech. Tyto knihovny, navržené v jazyce Matlabu, nabízejí předzpracované specializované funkce, které je možno rozšiřovat, modifikovat, anebo jen čerpat informace z přehledně dokumentovaných algoritmů. [1]



Obr.1 Základní princip práce s programem Matlab-Simulink

2.2 Simulink

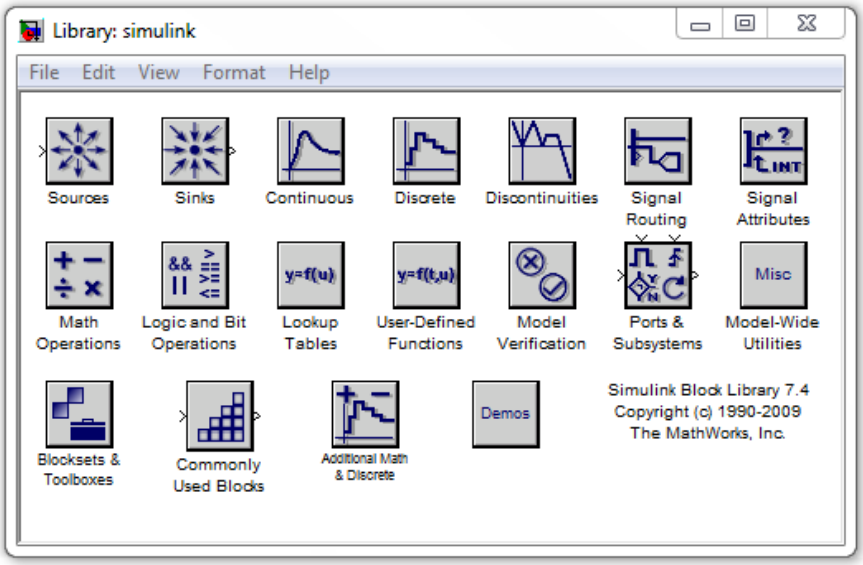
Simulink je nadstavba programu Matlab pro simulaci a modelování dynamických systémů, který využívá algoritmy Matlabu pro numerické řešení nelineárních diferenciálních rovnic. Poskytuje uživateli možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat a rovnic.

Kromě standardních úloh dovoluje Simulink rychle a přesně simulovat i rozsáhlé systémy s efektivním využitím paměti počítače. Pomocí Simulinku a jeho grafického editoru lze vytvářet modely lineárních, nelineárních, v čase diskrétních nebo spojitých systémů pouhým přesouváním funkčních bloků myší. Samozřejmostí je otevřená architektura, která dovoluje uživateli vytvářet si vlastní funkční bloky a rozšiřovat již tak bohatou knihovnu Simulinku. Simulink, stejně jako Matlab, dovoluje připojovat funkce napsané uživateli v jazyce C. Vynikající grafické možnosti Simulinku je možné přímo využít k tvorbě dokumentace. Mezi neocenitelné vlastnosti Simulinku patří nezávislost uživatelského rozhraní na počítačové platformě. Přenositelnost modelů a schémat mezi různými typy počítačů umožňuje vytvářet rozsáhlé modely, které vyžadují spolupráci většího kolektivu řešitelů na různých úrovních. [1]

3. Standardní knihovny Simulinku

Jak již bylo řečeno v úvodu tohoto textu, nadstavbové prostředí Simulink je určeno především k modelování a simulaci dynamických systémů. Toto prostředí využívá grafických možností operačního systému Windows. Způsob vytváření vlastního modelu je tedy značně odlišný od práce se základním prostředím Matlabu. Modely uživatel definuje jednoduchým způsobem prostřednictvím blokových schémat, která jsou podobná zapojením pro analogový počítač. Pro efektivní práci v Simulinku je nutná základní znalost Matlabu. Nezbytné jsou ale také alespoň elementární znalosti principů matematického modelování dynamických systémů.[2]

Ke knihovnám Simulinku lze přistupovat prostřednictvím okna Simulink Library Browser.



Obr.2 Základní okno knihoven v Simulinku

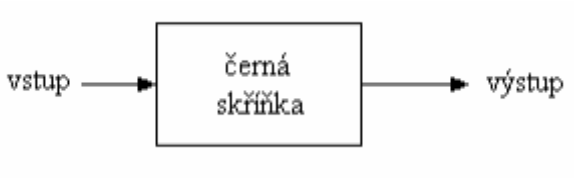
V mé práci se zaměřím na analýzu vlastností těch knihoven a jejich bloků vhodných pro simulaci řízení elektrických pohonů, což jsou převážně standardní knihovny Continuous a Discrete.

3.1 Analýza knihovny Continuous a jejich vybraných bloků

Knihovna Continuous slouží ke spojitému popisu dynamických systémů.

Systém si lze představit jako černou skříňku, jeho vlastnosti lze popsat pomocí reakcí výstupů na vstupní signály.Dynamické vlastnosti systémů jsou určeny právě vztahy mezi výstupními a vstupními veličinami.Jaké děje probíhají uvnitř systému z hlediska popisu vlastností systému u této metody

nejsou podstatné.Vnější popis je výhodný z hlediska využití analogie mezi systémy z různých oblastí.[3]

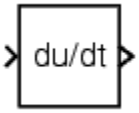


Obr.3 Systém jako černá skříňka

Dynamické vlastnosti spojitých systémů lze popsat buď v časové nebo frekvenční oblasti následujícími způsoby.

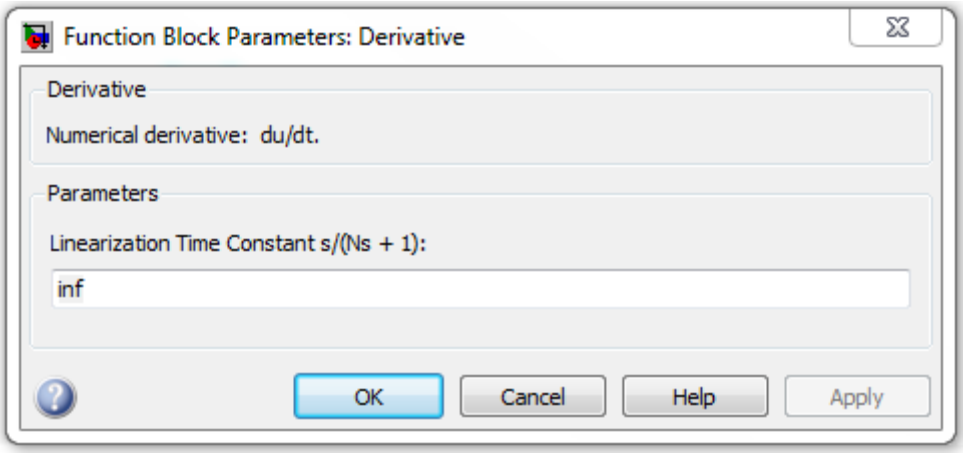
- diferenciální rovnicí
- operátorovým přenosem (Laplaceova transformace)
- frekvenčním přenosem (Fouierova transformace)
- frekvenční charakteristikou

3.1.1 Blok Derivative



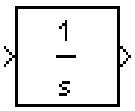
Obr.4 Schématická značka bloku Derivative

Blok Derivative slouží k numerické derivaci. Je známou skutečností, že matematicky jasně definovaná derivace je obtížně numericky aproximovatelná.Výstupem derivátoru v daném čase je hodnota derivace vstupního signálu v tomto čase (derivace v bodě).Blok derivace umožňuje změnu konstanty N (časová konstanta), implicitní hodnota je Inf (nekonečno,přesněji aritmetická reprezentace kladného nekonečna podle standardu IEEE). Hodnota Inf reprezentuje linearizaci v 0.



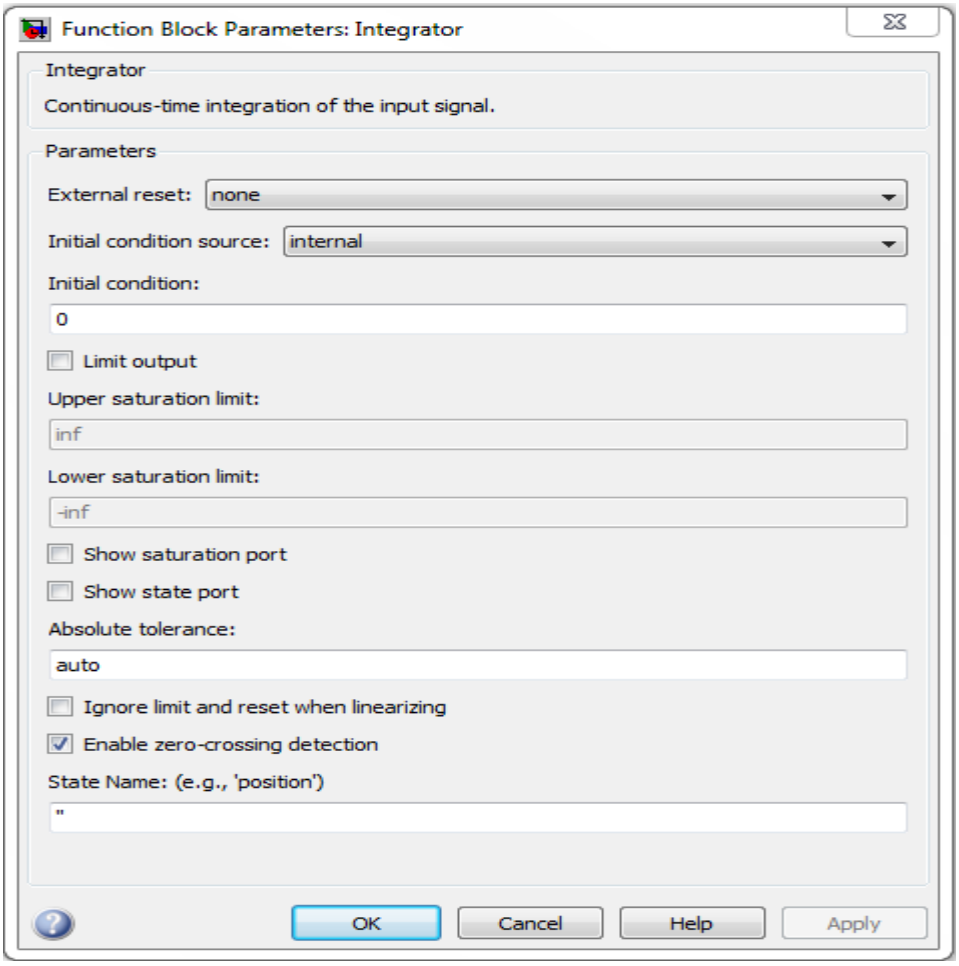
Obr.5 Okno pro nastavení parametrů bloku Derivative

3.1.2 Blok Integrator



Obr.6 Schématická značka bloku Integrator

Blok Integrator umožňuje zadání počátečních podmínek a limitaci výstupu (saturaci). Obojí je možno zadat jednak přímo hodnotou parametru, ale také pomocí dalších vstupních signálů. Blok dovoluje i externí reset (nastavení do počátečního stavu) na základě změny řídicího logického signálu (na náběžnou, sestupnou nebo obě hrany). Další možností je definování stavového výstupu (state portu), který se využívá právě ve spojitosti s externím resetem integrátoru a poskytuje hodnotu, která by byla na výstupu, pokud by nedošlo k resetu.



Obr.7 Okno pro nastavení parametrů bloku Integrator

Popis možností nastavení bloku:

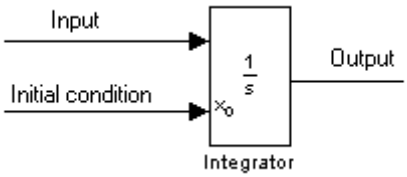
External reset – obnovuje stavy počátečních podmínek až se spouštěcí impuls vyskytne v obnovovacím signálu

- *none* - je to defaultní stav, kdy není obnoven počáteční stav
- *rising* - obnoví stav, kdy se resetovací signál zvedne z nuly na kladnou hodnotu, nebo z negativní na pozitivní hodnotu
- *falling* - obnoví stav, kdy resetovací signál spadne z kladné hodnoty na nulu nebo z pozitivní na negativní hodnotu
- *either* - obnoví stav, kdy resetovací signál změní z nuly na nenulovou hodnotu nebo změní znaménko
- *level* - obnoví stav, kdy se resetovací signál změní z nenulového současného kroku nebo z nenulového předchozího kroku na nulový současný časový krok
- *level hold* - obnoví stav, kdy resetovací signál je nenulový v současném časovém kroku

Initial condition source – získaká počáteční podmínky stavů

- *internal* - získaká počáteční podmínky stavů z počátečního stavu parametru
- *external* - získaká počáteční podmínky stavů z externího bloku

Initial condition – specifikuje stavy počátečních podmínek, defaultní hodnota je nastavena na nulu.

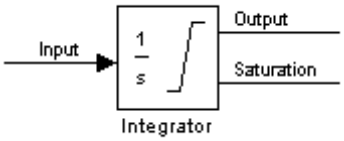


Obr.8 Schématický blok pro definování počáteční podmínky

Limit output – omezuje bloky výstupů na hodnoty mezi dolní a horní mezí saturace.Defaultně je hodnota vypnuta (Off), což značí nemožnost změn mezí saturace, kdybychom chtěli meze saturace změnit tak zaškrtneme možnost On

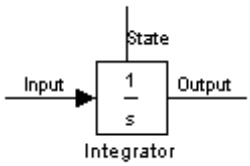
- *Upper saturation limit* – horní hodnota saturace
- *Lower saturation limit* – dolní hodnota saturace

Show saturation port – přidává saturační výstupní port bloku.Defaultně není tato možnost povolena, zašktnutím ji povolíme



Obr.9 Schématický blok pro definování omezení (saturace)

Show state port – přidává výstupní port bloku pro blokové stavy. Defaultně není tato možnost povolena, zašktnutím ji povolíme



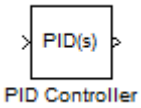
Obr.10 Schématický blok pro definování stavového portu

Absolute tolerance – specifikuje absolutní tolerance pro výpočet výstupního bloku.Můžeme nastavit automatickou nebo numerickou hodnotu

Ignore limit and reset when linearizing – způsobuje linearizaci pomocí příkazu, které upravují blok, aby nebyly limity na výstupu bez ohledu na nastavení bloků resetovány a omezeny nastavením. Defaultní hodnota je Off

Enable zero – crossing detection – Defaultní hodnota je nastavena na On, znamená to, že rozpozná a vezme časový krok pro nějaké následující události - reset nebo vstupní či výstupní horní nebo dolní saturační stav

3.1.3 Blok PID controller



Obr.11 Schématická značka bloku PID controller

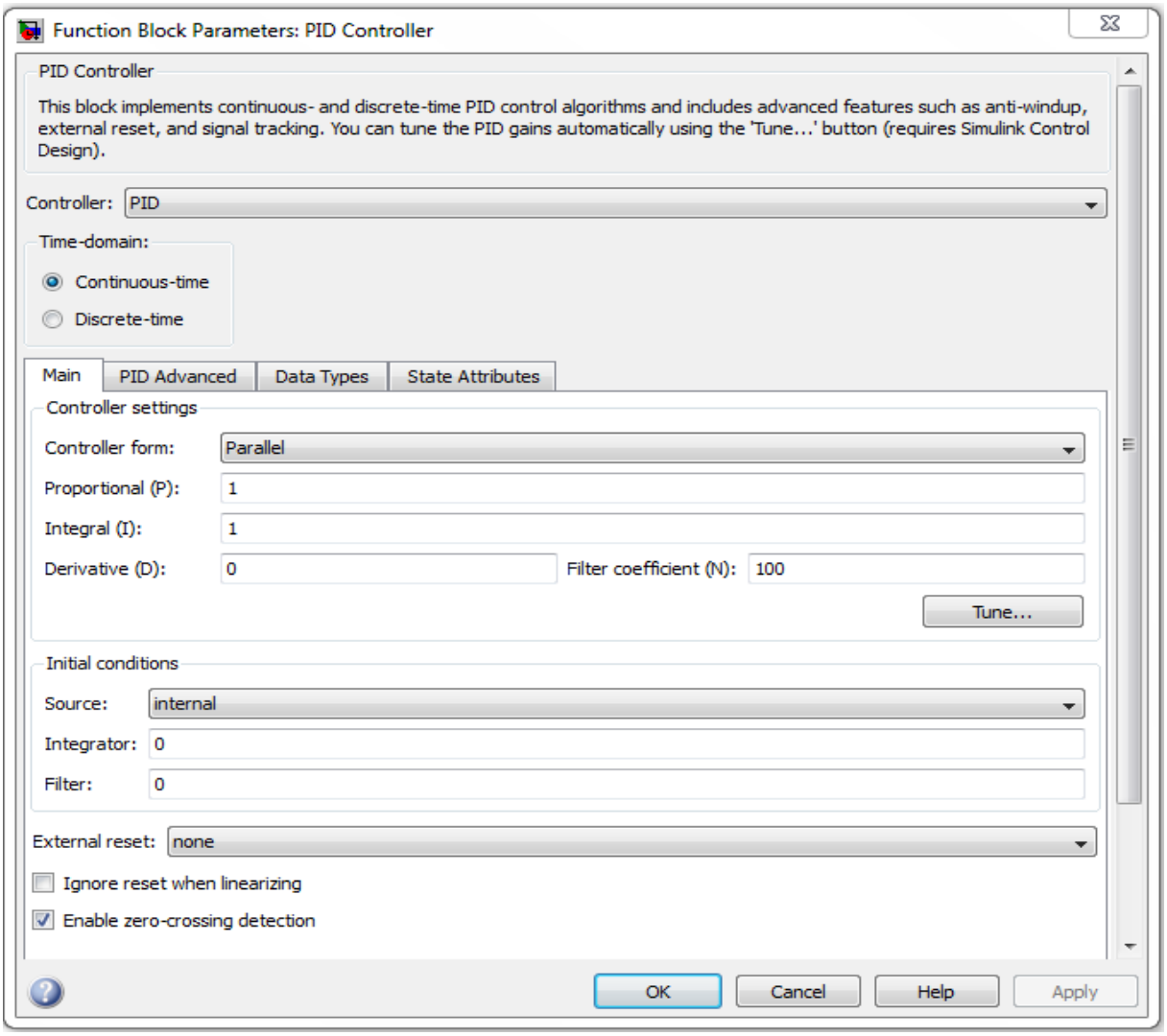
Výstupní blok PID regulátoru je součtem vstupního integračního a derivačního signálu.Důležité jsou proporcionální, integrační a derivační zesilovací parametry.Tento blok realizuje spojitý a diskrétní PID algoritmy včetně funkcí jako např. sledování signálu nebo externí nulování. PID zesílení můžeme automaticky nastavit pomocí tlačítka Tune, který vyžaduje Simulink Control Design.

V okně pro nastavení PID regulátoru si nejprve můžeme zvolit typ regulátoru: PID, PI, PD, P, I, kde P-člen značí proporcionální regulátor, I- člen je integrační a D-člen je derivační regulátor.Analýza v časové oblasti může být spojitá popřípadě diskrétní.

Tento blok má rozdělení dialogového okna do čtyř záložek:

- Main
- PID Advanced
- Data Types
- State Attributes

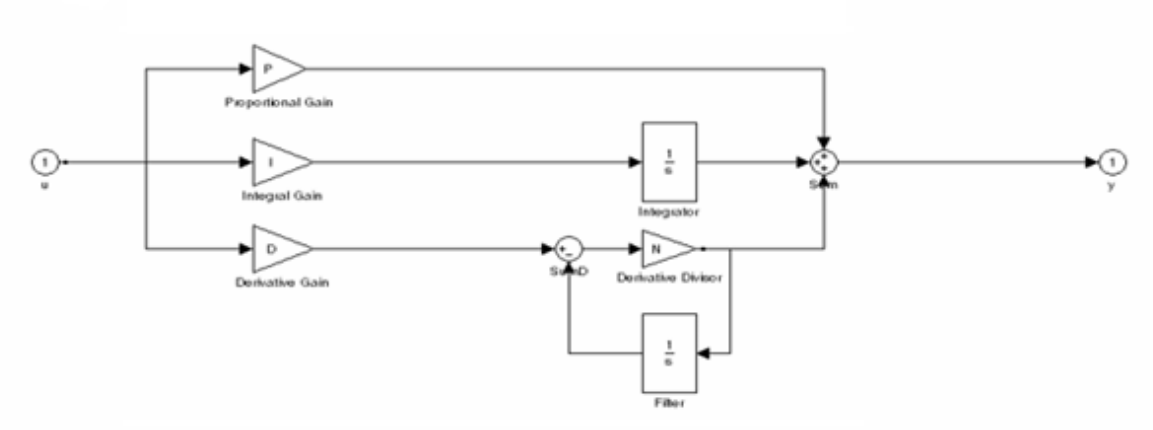
Nyní si rozebereme záložku ***Main,*** kde se dají nastavit základní možnosti PID regulátoru



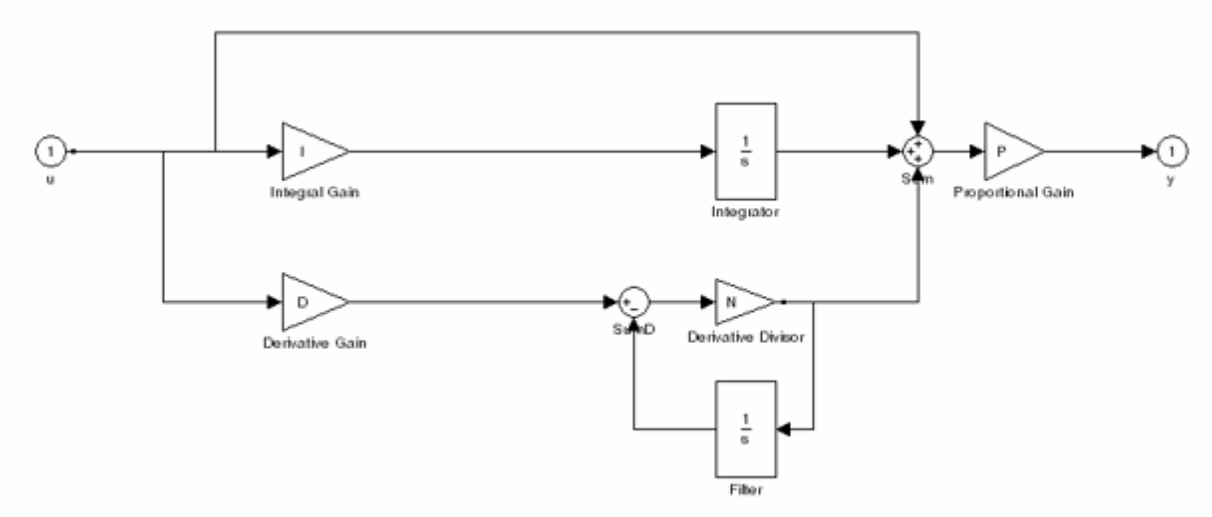
Obr.12 Okno pro nastavení parametrů bloku PID Controller

Controller form (forma regulátoru) - můžeme nastavit paralelní (nastaveno jako defaultní) nebo ideální.Paralelní vybírá formu regulátoru, která je součtem proporcionálních, integračních a derivačních vlivů.Všechnny tři složky (P, I, D) mohou být nastavovány nezávisle (obr.13)

Naproti tomu u ideální formy působí proporcionální složka až na součet složek integračních a derivačních (obr.14).



Obr.13 Blokové schéma paralelní formy PID regulátoru



Obr.14 Blokové schéma ideální formy PID regulátoru

Proporcional (P) – vhodné pro PID, PD, PI a P regulátory. Specifikuje proporcionální zesílení P.Defaultně je nastavena hodnota rovna jedné.Pro paralelní formy PID regulátoru je nezávislý na integračním a derivačním vlivu.Pro ideální formy PID regulátoru působí proporcionální člen na integrační i derivační členy.

Integral (I) – vhodné pro PID, PI a I regulátory. Specifikuje integrační zesílení I. Defaultně je nastavena hodnota rovna jedné.

Derivative (D) – vhodné pro PID a PD regulátory. Specifikuje derivační zesílení D. Defaultně je nastavena hodnota rovna nule.

Filter coefficient (N) – tento koeficient je dostupný jen pro PID a PD regulátory. Specifikuje hodnotu derivační časové konstanty ve formě filtru, defaultní hodnota = 100.

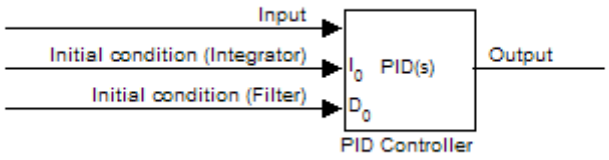
Initial condition source – (počáteční stav zdroje) můžeme nastavit internal (nastaveno defaultně) nebo external.

- *internal* (vnitřní) - Specifikuje integrační a filrační vstupní podmínky jednoznačně pomocí integračních a filtračních počátečních parametrů.

- *external* (vnější) – Specifikuje integrační a filtrační podmínky exterrně.

Integrator Initial condition – defaultně nastaven na nulu. Je k dispozici pouze při počáteční stavu zdroje nastaveným na internal.Specifikuje integrační počáteční hodnotu.

Filter Initial condition - defaultně nastaven na nulu. Je k dispozici pouze při počáteční stavu zdroje nastaveným na internal. Specifikuje filtrační počáteční hodnotu.



Obr.15 Schématický blok pro definování počátečních stavů

External reset – zvolí případ, kdy obnoví integrační a filtrační výstupy na počáteční podmínky, které se zadají do polí Integrator a Filter.

- *none* - nastaveno defaultně. Nenastavuje integrační a filtrační výstupy na počáteční podmínky.

- *rising* - resetuje výstupy, kdy resetovací signál má stoupající hranu.

- *falling* - resetuje výstupy, kdy resetovací signál má sestupnou hranu.

- *either* - resetuje výstupy, kdy resetovací signál buď stoupá nebo klesá.

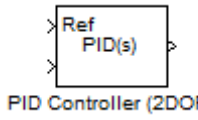
- *level* - obnoví a drží výstupy na počátečních podmínkách zatímco resetovací signál je nenulový.

Ignore reset when linearizing (ignoruje reset při linearizaci) - simulinkové linearizační příkazy neignorují stavy korespondující reset mechanismu.

Enable zero-crossing detection - simulinkové linearizační příkazy ignorují stavy korespondující reset mechanismu.

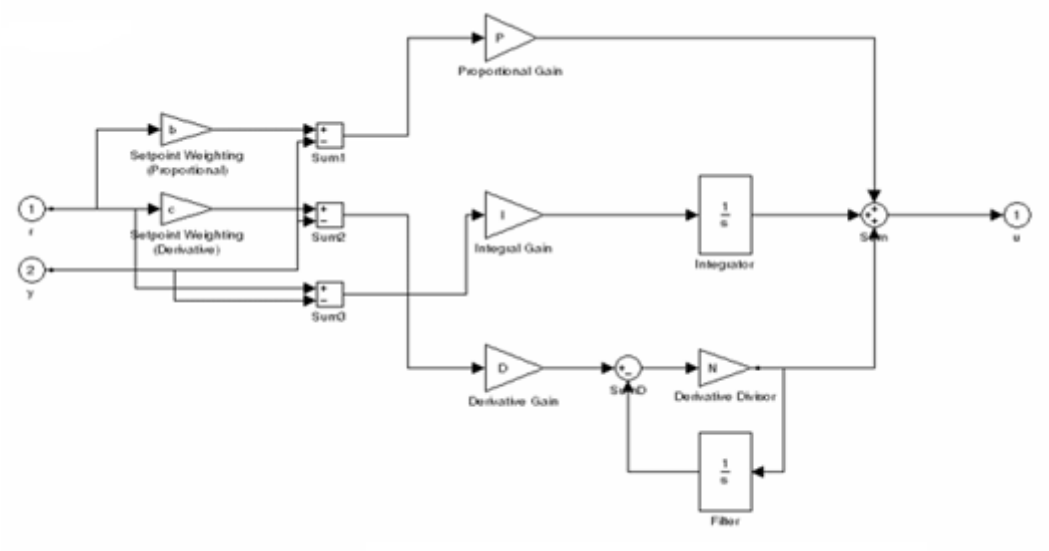
Nastavení PID regulátoru je poměrně rozsáhlé. Popis všech možností nastavení v dalších záložkách, které jsem zmínil výše jsou uvedeny v příloze této bakalářské práce.[P1]

3.1.4 Blok PID controller (2DOF)

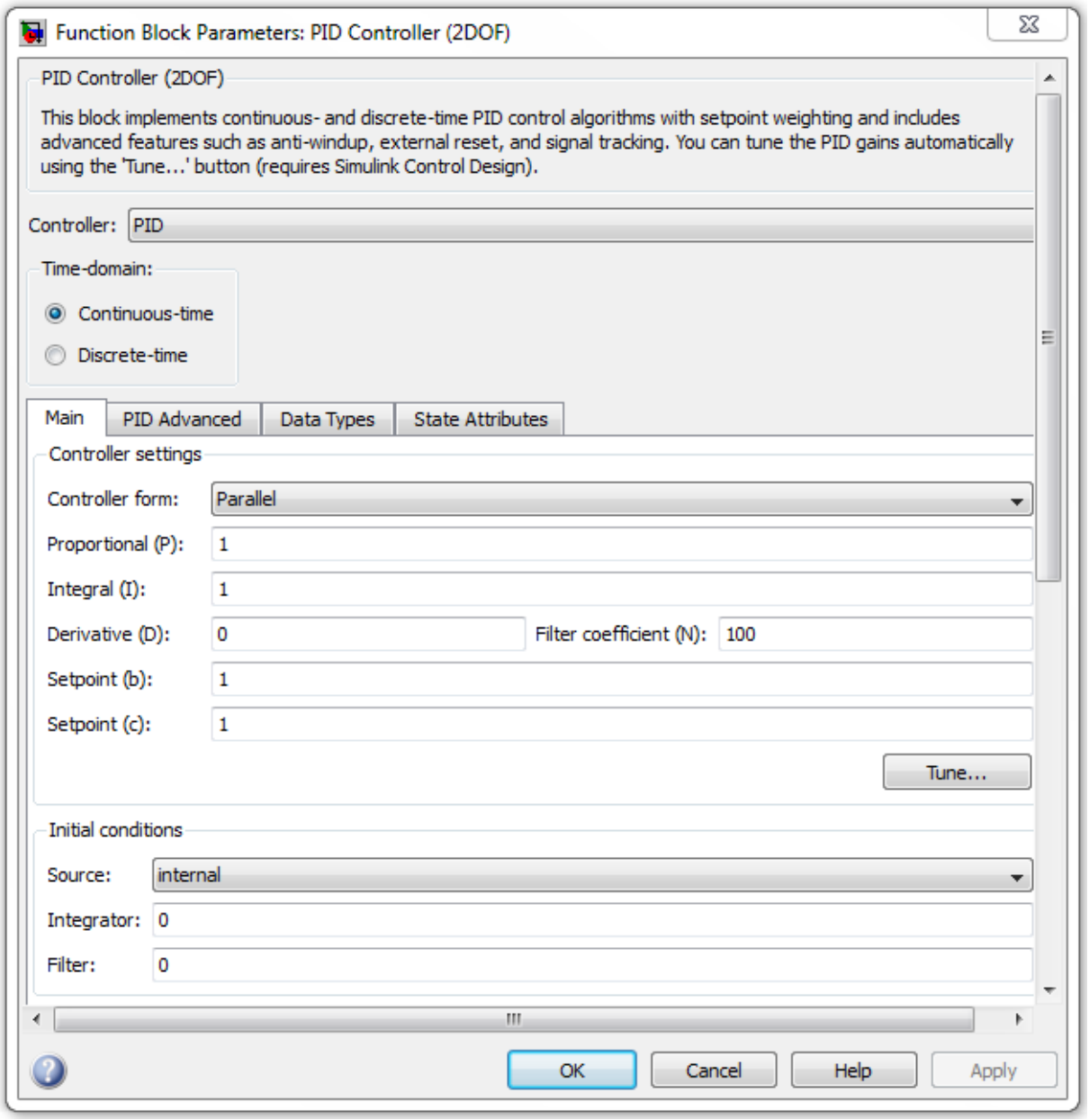


Obr.16 Schématická značka bloku PID controller (2DOF)

Tento blok umožňuje nastavit dva stupně volnosti regulátoru PID nebo PI, PD v programu Simulink.Ve standardní regulační smyčce PID regulátor generuje akční veličinu *u* na základě regulační odchylky *e*. Tento regulátor neumožňuje seřídit parametry zvlášť pro optimální potlačení poruchy a také pro optimalizaci odezvy uzavřené smyčky na skokovou změnu požadované hodnoty. Regulátor PID (2DOF) umožňuje dosáhnout žádané sledování a dobré potlačení vlivu poruchových veličin na základě rozdílu mezi referenčním signálem a měřeným výstupem. Blok vypočítává rozdílný signál pro každý z proporcionální, derivační a integrační činnosti podle požadované hodnoty vah, které zadáme. Výstupní blok je součtem příslušných rozdílových signálů.



Obr.17 Schématický blok PID regulátoru (2DOF)



Obr.18 Okno pro nastavení parametrů bloku PID Controller (2DOF)

V okně pro zadávání parametrů si můžeme všimnout, že kromě parametrů pro proporcionální, integrační a derivační složku můžeme ještě nastavit hodnoty setpoint (b) , setpoint (c).

- *Setpoint (b)* – váha žádané veličiny proporcionální složky.
- *Setpoint (c)* – váha žádané veličiny derivační složky.

Váhy b i váha c se dají měnit v rozmezí od 0 do 1. Pokud jsou nastaveny na hodnotu 1 tak se regulátor chová jako regulátor s jedním stupněm volnosti.[6]

Pokud jde o další možnosti nastavení toho bloku tak jsou stejné jako u standardního PID regulátoru popsaného výše a v příloze této bakalářské práce. [P1]

3.2 Analýza knihovny Discrete a jejich vybraných bloků

Knihovna Discrete slouží k diskrétnímu popisu dynamických systémů. Většina bloků je víceméně ekvivalentní blokům z knihovny Continuous, např. bloky Discrete-Time Integrator, Discrete Transfer Fcn resp. Discrete Filter, Discrete State-Space.Knihovna obsahuje také bloky tvarovače nultého řádu (Zero-Order Hold) a prvního řádu (First-Order Hold). Užitečný je i blok Memory, který představuje zpoždění vstupního signálu o jeden simulační krok a lze jej využít např. k odstranění algebraické smyčky [2].

Diskrétní systémy mohou být diskrétní ve smyslu amplitudy jednotlivých veličin nebo ve smyslu časového průběhu.

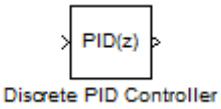
Diskrétní dynamické systémy pak lze definovat tak, že všechny definované veličiny buď existují nebo jsou měřitelné (měřené) pouze v oddělených časových okamžicích. Pokud intervaly mezi těmito okamžiky jsou konstantní, mluvíme o diskrétních systémech s pravidelným (periodickým) vzorkováním.
První diskrétní systémy se používaly ve 30.letech minulého století. Zjistilo se, že jsou vhodné pro řízení systémů s dopravním zpožděním.Další rozvoj byl ovlivněn použitím měřících metod, kde snímače dávaly informaci v diskrétní formě (radiolokátory, čidla s číslicovým výstupem, chemické analýzy.) Jejich hlavní rozvoj ovšem nastal až při nasazení počítačů ve funkci regulátorů v rámci automatizace technologických procesů. I když většina řízených systemů je spojitá, regulační obvod s počítačem je nutno řešit jako diskrétní systém.

Dynamické vlastnosti diskrétních systémů, na rozdíl od spojitých systémů, lze popsat pouze v časové oblasti a to těmito způsoby.

- diferenční rovnicí
- diskrétním obrazovým přenosem
- impulsní charakteristikou
- přechodovou charakteristikou
- rozložením nul a pólů diskrétního přenosu

V některých případech se používá i popis v takzvané pseudo-frekvenční oblasti, kdy se provede transformace diskrétního systému do roviny obdobné rovině „s“ pro spojité systémy.Takto se dají porovnat vlastnosti diskrétních a spojitých systémů ve frekvenční oblasti.[3]

3.2.1 Diskrétní blok PID controller



Obr.19 Schématická značka diskrétního bloku PID controlleru

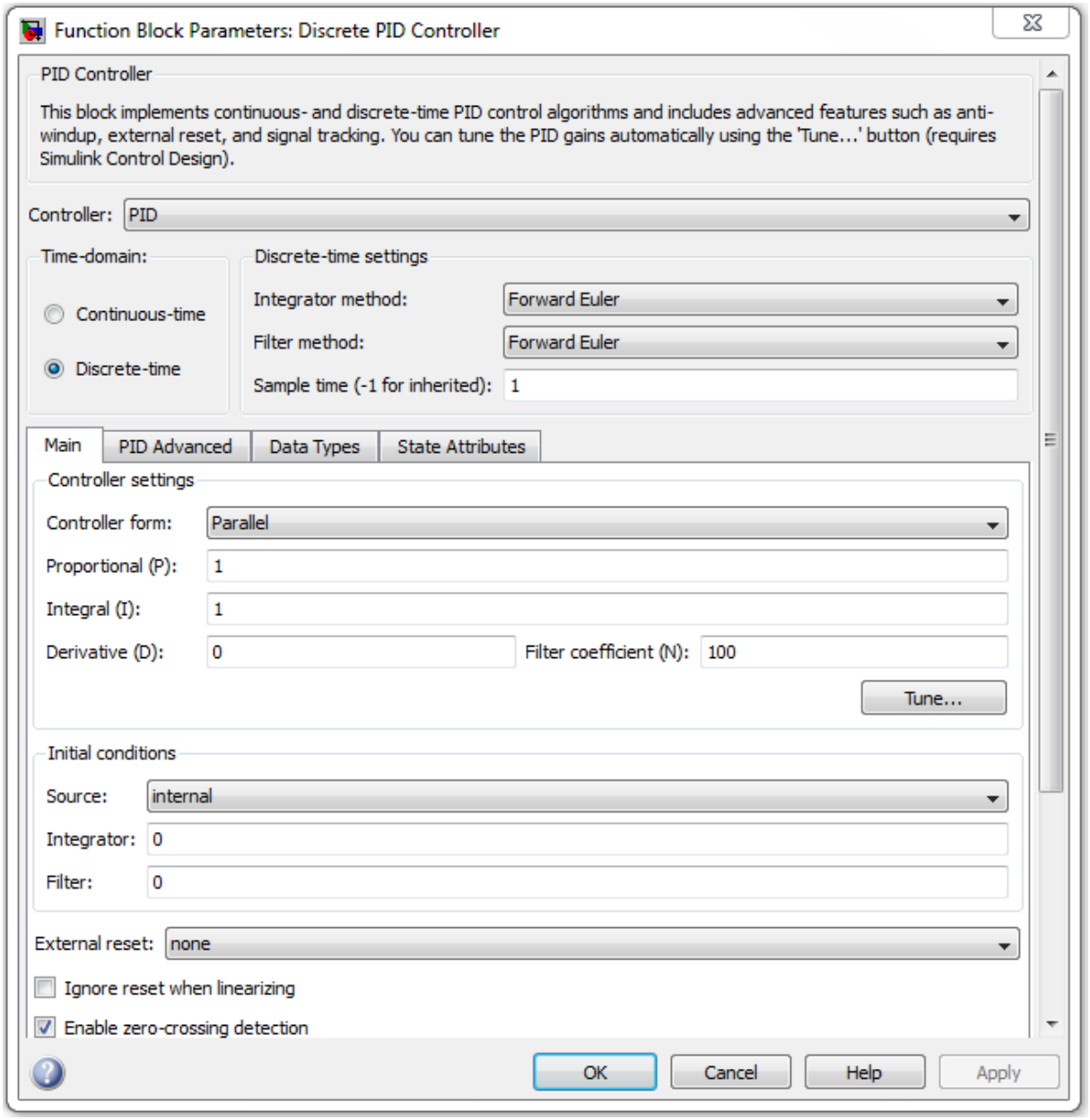
Tento blok implementuje diskrétní algoritmy PID regulátoru a obsahuje pokročilé funkce jako je anti-windup, external reset and signal tracking.

Jak lze vidět z okna pro nastavení parametrů tohoto bloku, tak jsme schopni nastavit stejné parametry jako u standardního PID controlleru.

Vyjímkou je možnost nastavení Discrete-time settings (diskrétního nastavení času).

- Integrační metoda
  - Forward Euler (přední Eulerova metoda)
  - Backward Euler (zpětná Eulerova metoda)
  - Trapezoidal (lichoběžníková metoda)

- Fitrační metoda
  - Forward Euler
  - Backward Euler
  - Trapezoidal



Obr.20 Okno pro nastavení parametrů diskrétního bloku PID Controller

Jinak můžeme říct, že možnosti nastavení jsou totožné jako u standardního PID regulátoru, kromě záložky **State Attributes**, kde je možné kromě zadání názvu stavu (integračního a filtračního) také nastavit *Real-Time Workshop storage class* (uložení hodnoty v reálném čase).

- *Auto* – je vhodné pro uložení stavů, které nemusíme připojit k externím hodnotám
- *ExportedGlobal* – stav je uložen v globální proměnné
- *ImportedExtern* – deklaruje stav jako externí proměnná
- *ImportedExternPointer* – deklaruje stav jako externí ukazatel

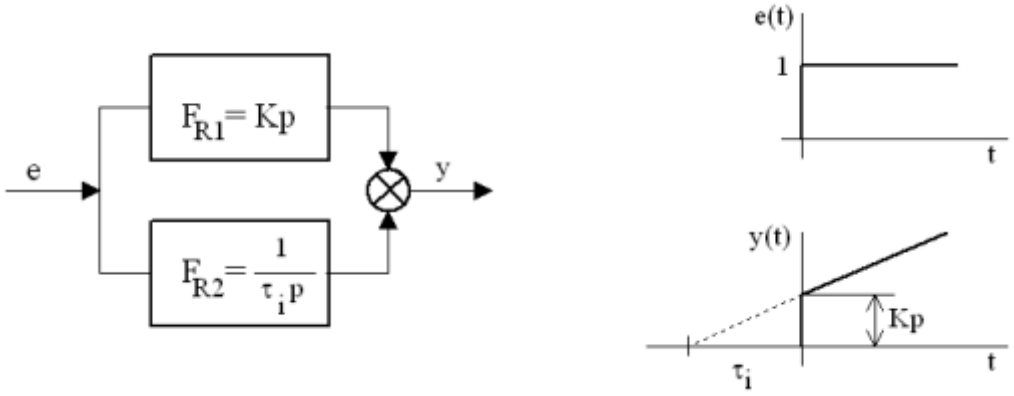
4. Simulace PI regulátoru v prostředí programu Simulink

Pro mé simulace jsem si vybral simulaci PI regulátoru s omezením integrační složky, ve které využiji pro simulaci některé bloky, které jsou popsány v kapitole 3.

Simulace PI regulátoru s omezením integrační složky je rozdělena na dvě části:

- omezením integrační složky pomocí bloku Fcn
- omezením integrační složky pomocí metody Back – calculation („zpětným přepočtem“)

PI regulátor patří k nejpoužívanějším typům regulátorů v technice elektrických pohonů. Tento regulátor vznikne paralelním spojením proporcionálního a integračního členu.

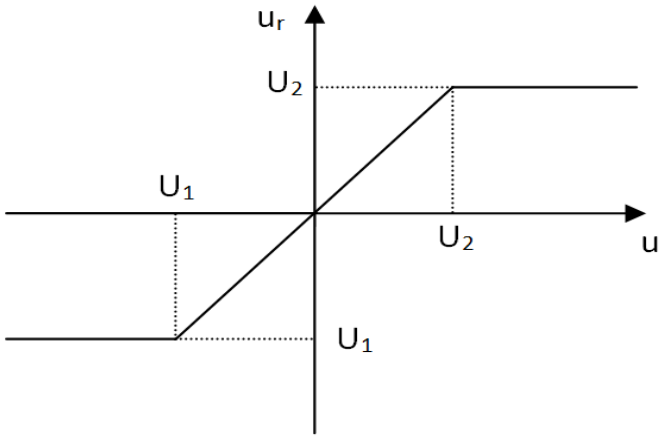


Obr.21 PI regulátor a jeho přechodová charakteristika

Přenos PI regulátoru je dán následným vztahem:

$$F_R(p) = F_{R1}(p) + F_{R2}(p) = K_P + \frac{1}{\tau_i p} = K_P \cdot \left(1 + \frac{1}{\tau_i p}\right) \quad (4\text{-}1)$$

kde $\tau_R = K_p \cdot \tau_i$ je časová konstanta, τ_i je časová konstanta integračního členu a K_p je zesílení proporcionálního členu.
Výstup regulátoru je tedy dán součtem proporcionální a integrační složky vstupního signálu. Nevýhodou tohoto jednoduchého zapojení je vzájemná vazba mezi zesílením a časovou konstantou. Není tedy možné nastavit tyto složky nezávisle na sobě.Nedostatek můžeme odstranit paralelním zapojením P regulátoru a I regulátoru zvlášť a následně pomocí sumátoru sečíst složky P a I. V praxi je většinou nutné vybavit regulátor omezovačem, který upraví výstupní signál na hodnotu potřebnou pro další zpracování. Kromě toho je vhodné vybavit regulátor i tzv. vnitřním omezovačem, který omezuje potřebnou hodnotu na integračním členu. Důvodem je fakt, že při dosažení výstupního omezení nebo saturace integračního členu už neplatí, že výstupní napětí je dáno součtem P a I složky. Dochází tak ke zkreslení výstupního napětí a ke zpomalení regulace.

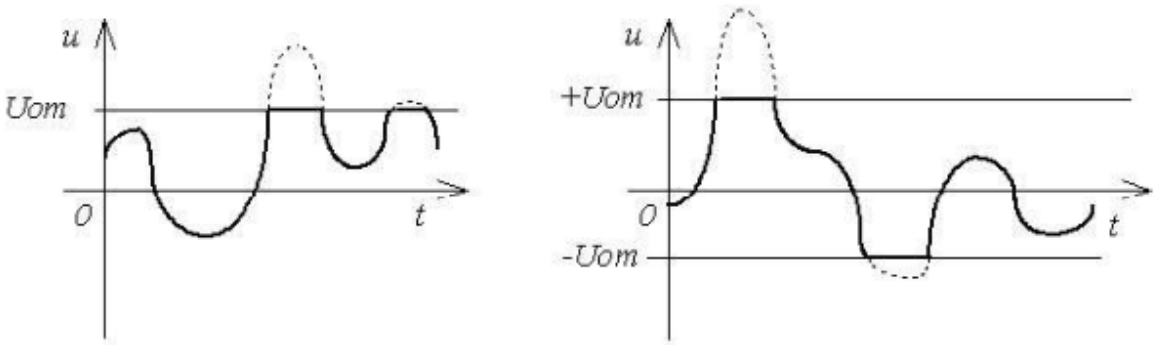


Obr.22 Obecná funkce omezovače

Z obr.22 můžeme vidět, že pokud má být hodnota u v mezích saturace musí platit:

$$U_1 \leq u \leq U_2 \tag{4-2}$$

Omezovače výstupní veličiny (amplitudy) se používají v regulačních obvodech například pro úpravu signálu na vstupu nebo výstupu regulátoru. Výsledkem je potom omezení regulované hodnoty (např. otáček, proudu, napětí) na určitou mez.Funkce omezovače je vyobrazena na obr.23, který znázorňuje časový průběh napětí při použití jednostranného a oboustranného omezovače.[4]



Obr.23 Funkce omezovače výstupní veličiny

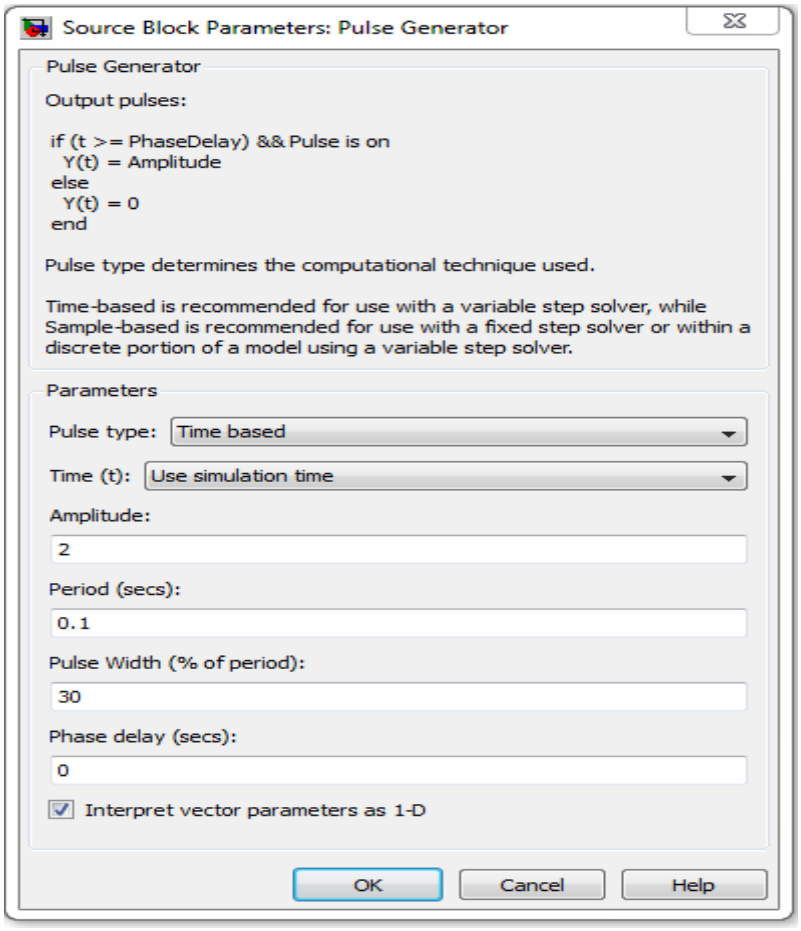
4.1 Simulace s omezením integrační složky pomocí bloku Fcn

Na obr.25 je simulační model PI regulátoru s vnitřní funkcí bloku Fcn. Nasimuloval jsem PI regulátor a zobrazil výstup z regulátoru, kde jsem omezil pomocí saturace výstup regulátoru a navíc jsem omezil integrační člen pomocí bloku Fcn na hodnotu [-10,10].

Blok Fcn je definován pomocí výrazu (u >= 10) + (u <= (-10)), čímž se definovalo požadované omezení integrační složky.Obr.27 je výstup z PI regulátoru omezen na požadované hodnotě.

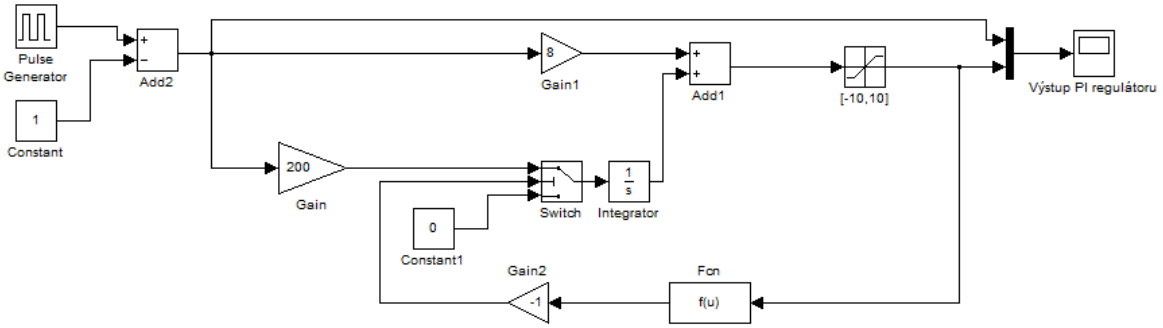
Vstupní signál je nastaven podle níže uvedeného schématu.Hodnoty jsou nastaveny následovně:

Amplituda = 2
Perioda = 0,1s
Šířka pulsu = 30% z hodnoty periody
Fázové zpoždění = 0s

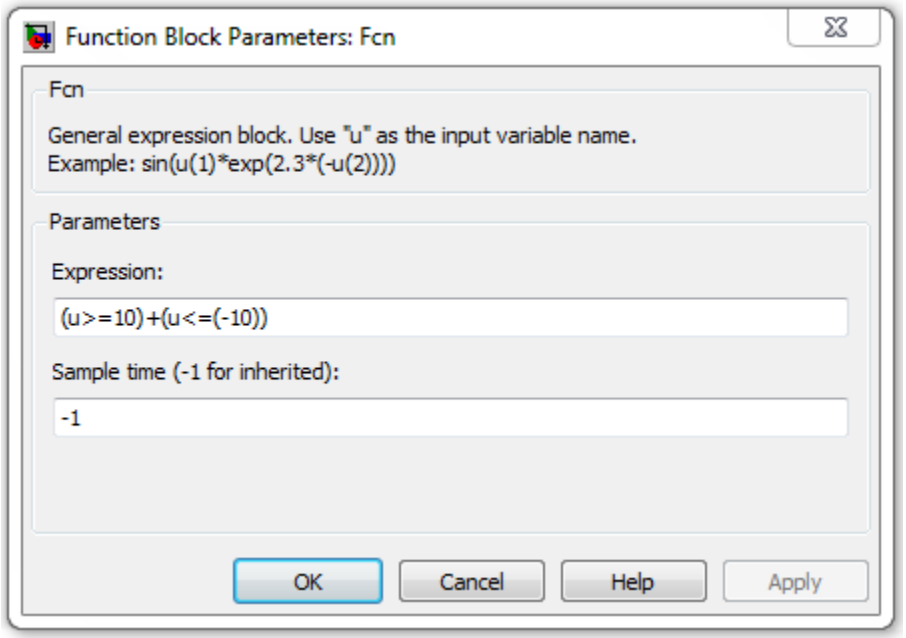


Obr.24 Hodnoty nastavení vstupního pulzního signálu

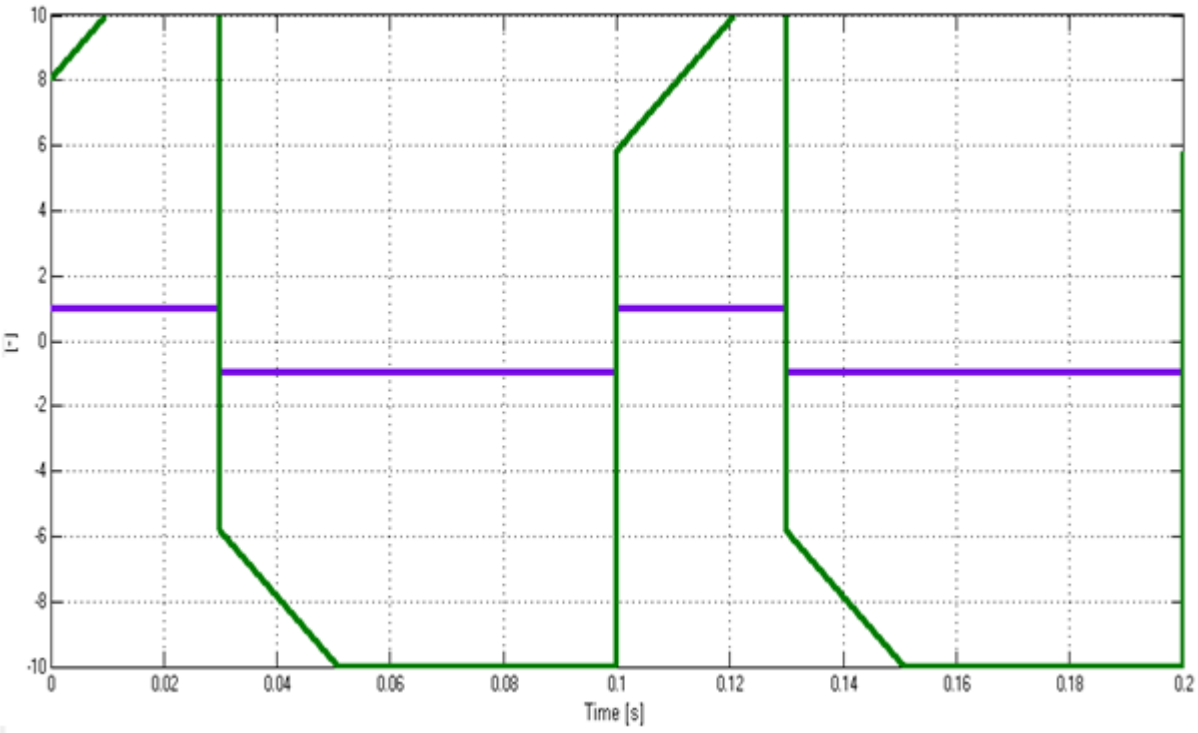
Hodnota zesílení P-složky je nastavena na hodnotu 8 a časová konstanta I-složky je rovna 5ms.



Obr.25 Simulační model PI regulátoru



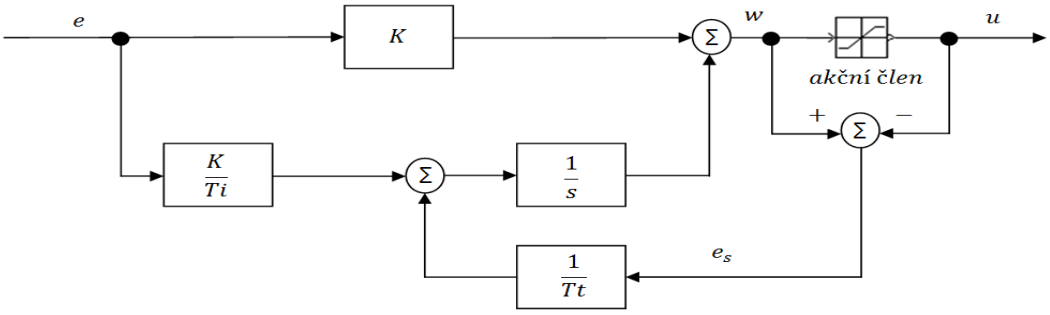
Obr.26 Nastavení bloku Fcn pro omezení integrační složky



Obr.27 Výstup z PI regulátoru

4.2 Simulace s omezením integrační složky pomocí metody Back – calculation

V tomto případě je omezení integrační složky provedeno metodou tzv. Back – calculation („zpětným přepočtem“). Základem metody je dynamický přepočet integračního vstupu pomocí časové konstanty Tt. Systém má dodatečnou zpětnou vazbu, která je vytvořena měřením výstupu akčního členu a odchylkovým signálem e_s vytvořeného z rozdílu mezi výstupem regulátoru u_r a výstupem akčního členu u. Odchylkový signál e_s je přiveden na vstup integrátoru přes zesílení 1/Tt.



Obr.28 Blokové schéma PI regulátoru s back-calculition

Když se nevyskytuje saturace, je tento signál nulový. Tudíž nebude mít žádný vliv na běžnou funkci. V případě saturace regulátoru se však signál e_s nebude rovnat nule a k integrátoru bude přiveden změněný signál. Běžná zpětná vazba systému je přerušena a výstup integrátoru je řízen směrem k vstupní hodnotě regulátoru. Tudíž vstup integrátoru se blíží k nule. Vstup integrátoru můžeme vyjádřit vztahem

$$\frac{1}{T_t} \cdot e_s + \frac{K}{T_i} \cdot e \qquad (4.2 - 1)$$

kde e je odchylka řízení. Z toho tedy plyne, že v ustáleném stavu platí

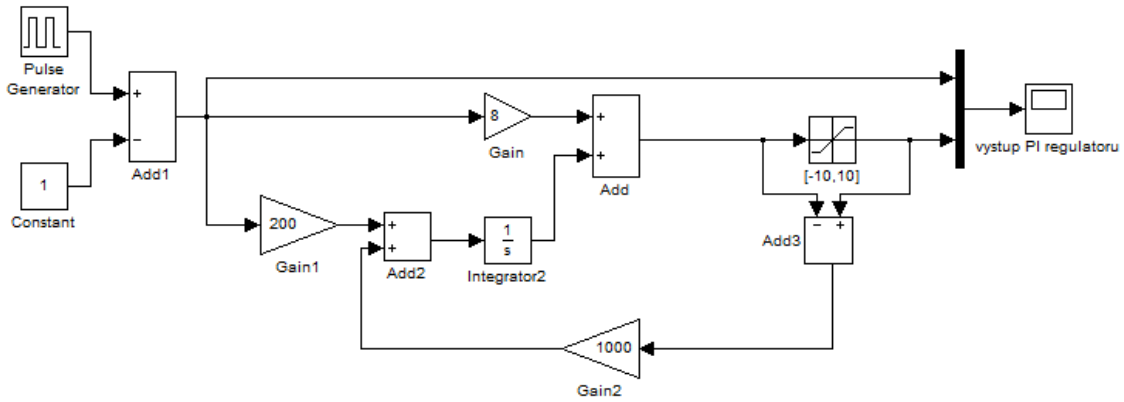
$$e_s = -\frac{KT_t}{T_i} \cdot e \qquad (4.2 - 2)$$

Zároveň také platí, že $e_s = u \pm u_r$, z čehož vyplývá

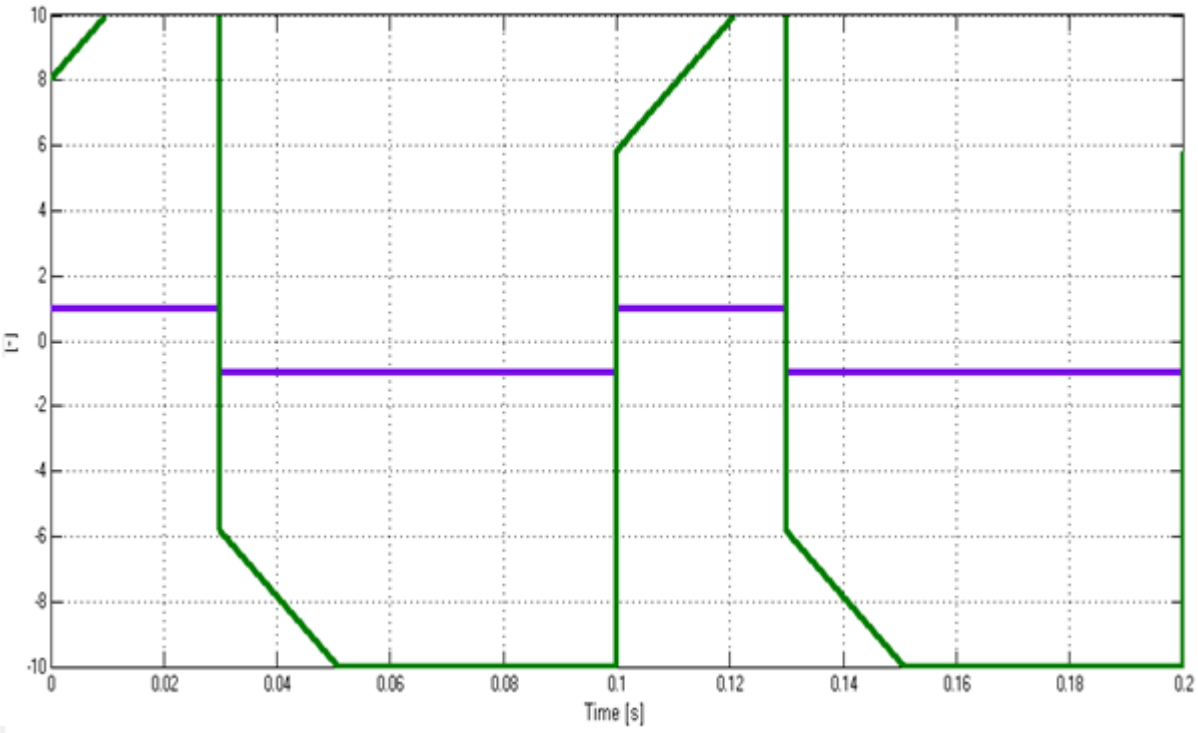
$$u_r = u_{lim} + \frac{KT_t}{T_i} \cdot e \qquad (4.2 - 3)$$

kde u_{lim} je saturační hodnota řízení. Takže e a u_{lim} mají stejné znaménko. Z toho plyne, že u_r je v absolutní hodnotě vždy větší než u_{lim}. Právě tato skutečnost zabraňuje vzniku integračního wind-up efektu.
Míra se kterou je výstup regulátoru seřízen, je řízena zpětnovazebním zesílením 1/Tt, kde Tt
může být chápána jako časová konstanta, která určuje jak rychle se integrátor přenastaví.
Nazýváme ji sledovací časová konstanta. Výhodnější je použití velmi malé hodnoty časové konstanty, protože integrační složka je pak rychleji změněna.[7]

Hodnoty zesílení P-členu a integrační časové konstanty jsou totožné jako u první metody, sledovací časová konstanta je nastavena na hodnotu 1ms.

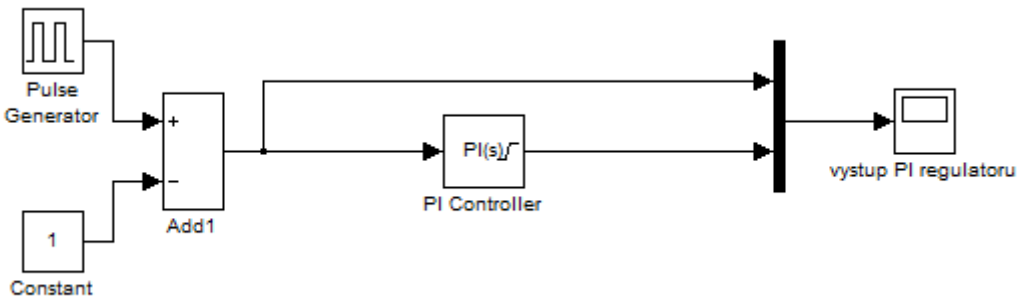


Obr.29 Simulační model PI regulátoru

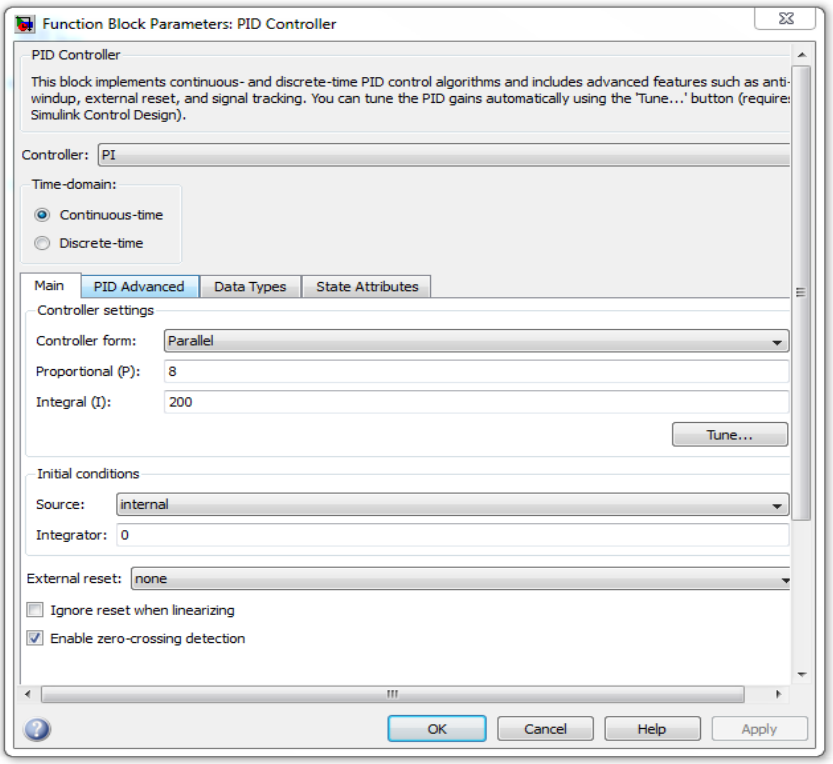


Obr.30 Výstup z PI regulátoru

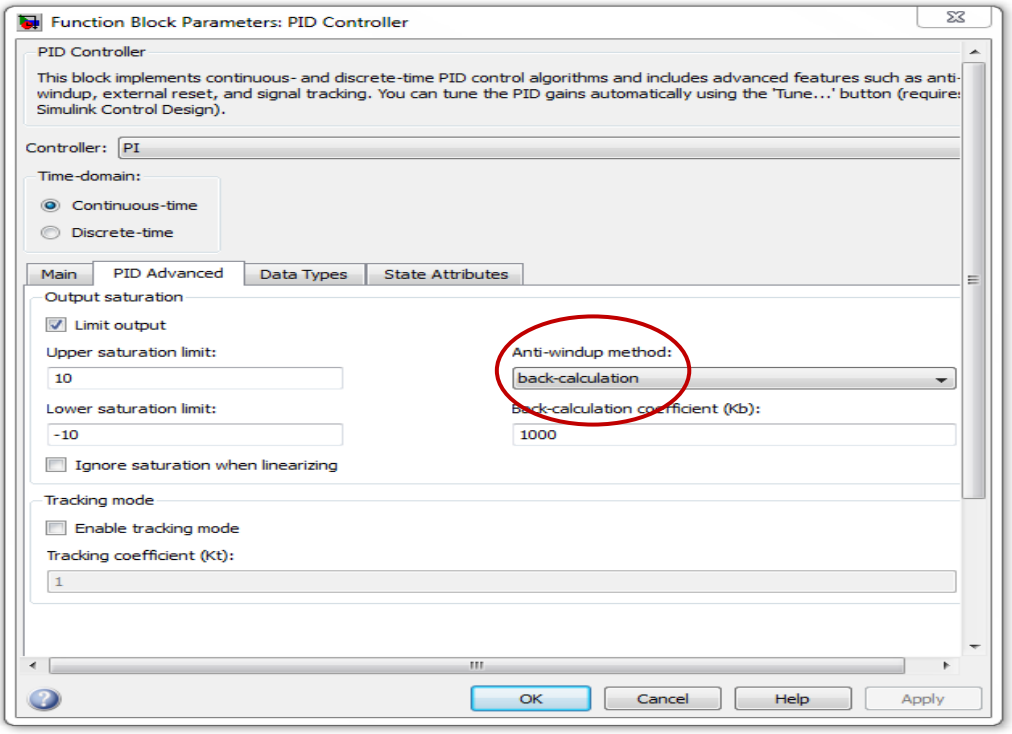
Simulace omezení integrační složky provedeno anti-wind up metodou tzv. Back – calculation („zpětným přepočtem“) pomocí hotového regulátoru ze simulinku, ve kterém jsem nastavil hodnoty zesílení proporcionální a integrační složky a také omezení integrační složky a nastavení anti-wind-up metody s časovou konstantou (koeficientem).



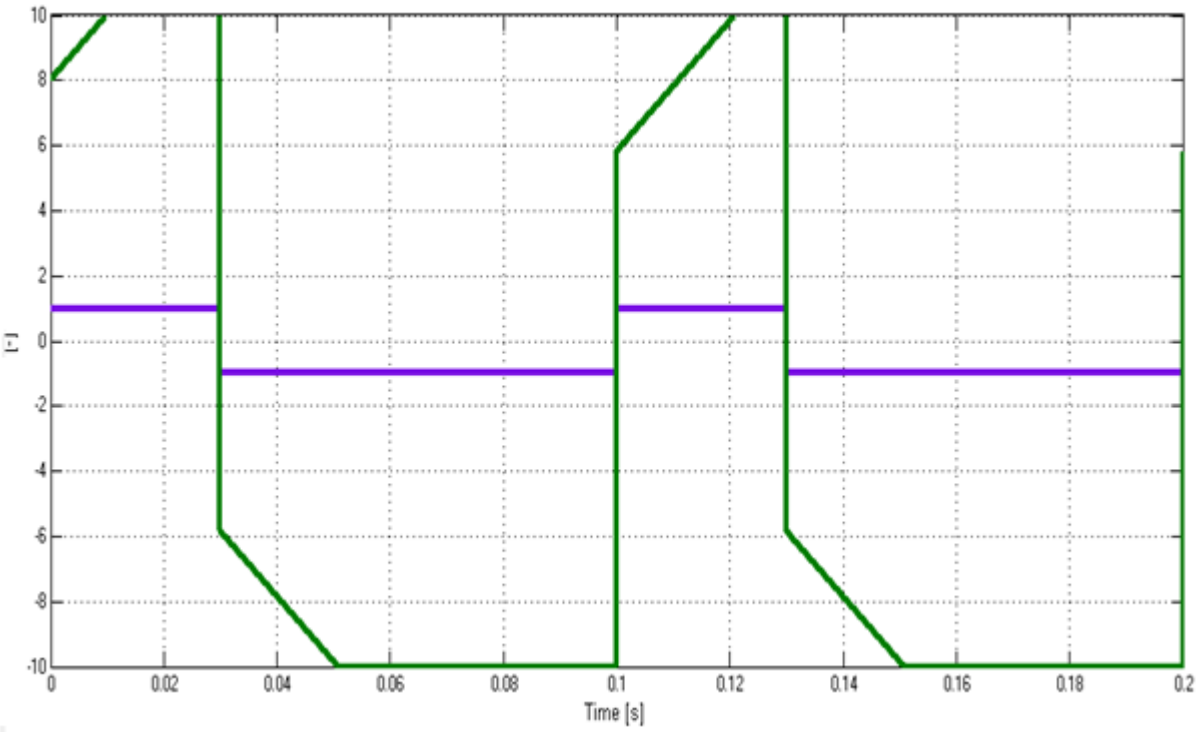
Obr.31 Simulační model PI regulátoru



Obr.32 Hodnoty nastavení vstupního PI regulátoru

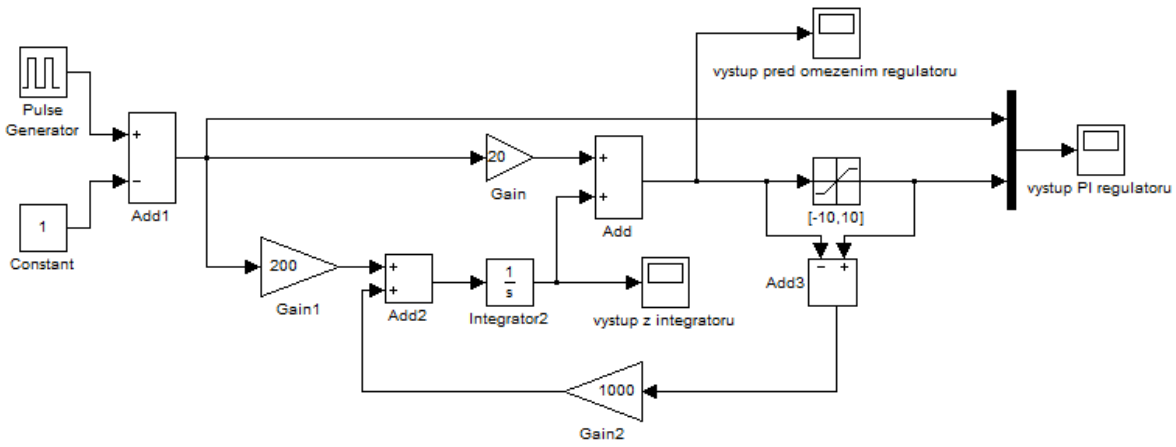


Obr.33 Hodnoty nastavení omezení pomocí Anti-windup metody

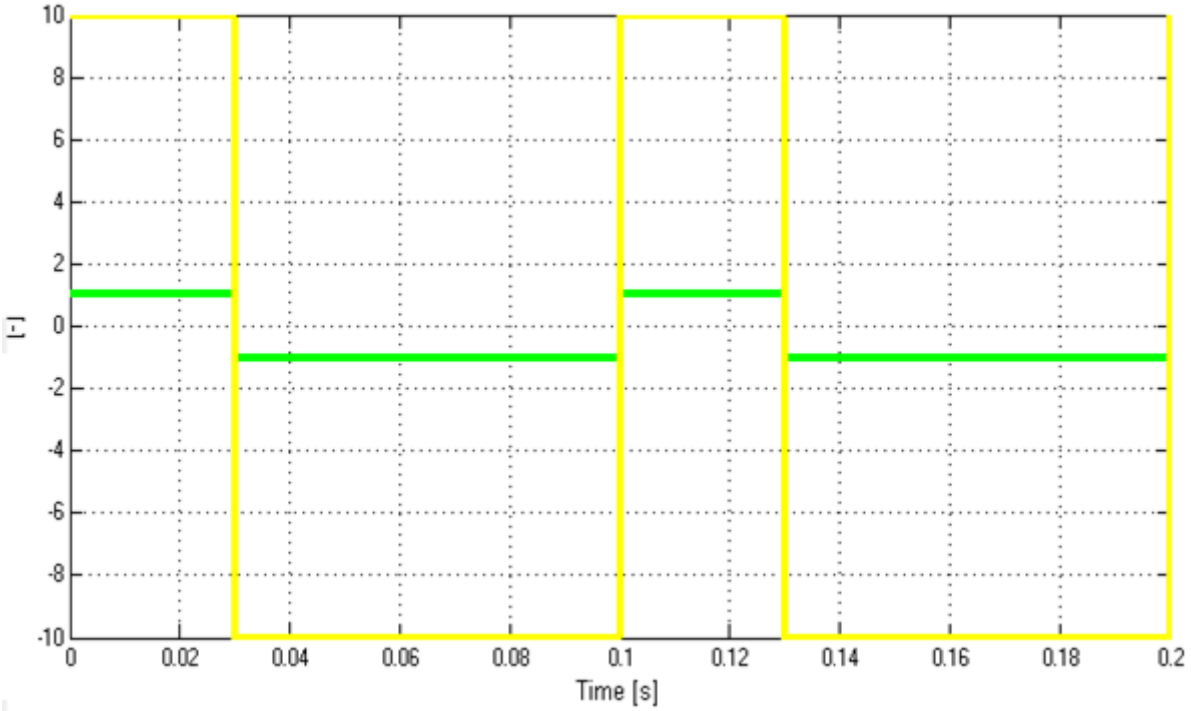


Obr.34 Výstup z PI regulátoru

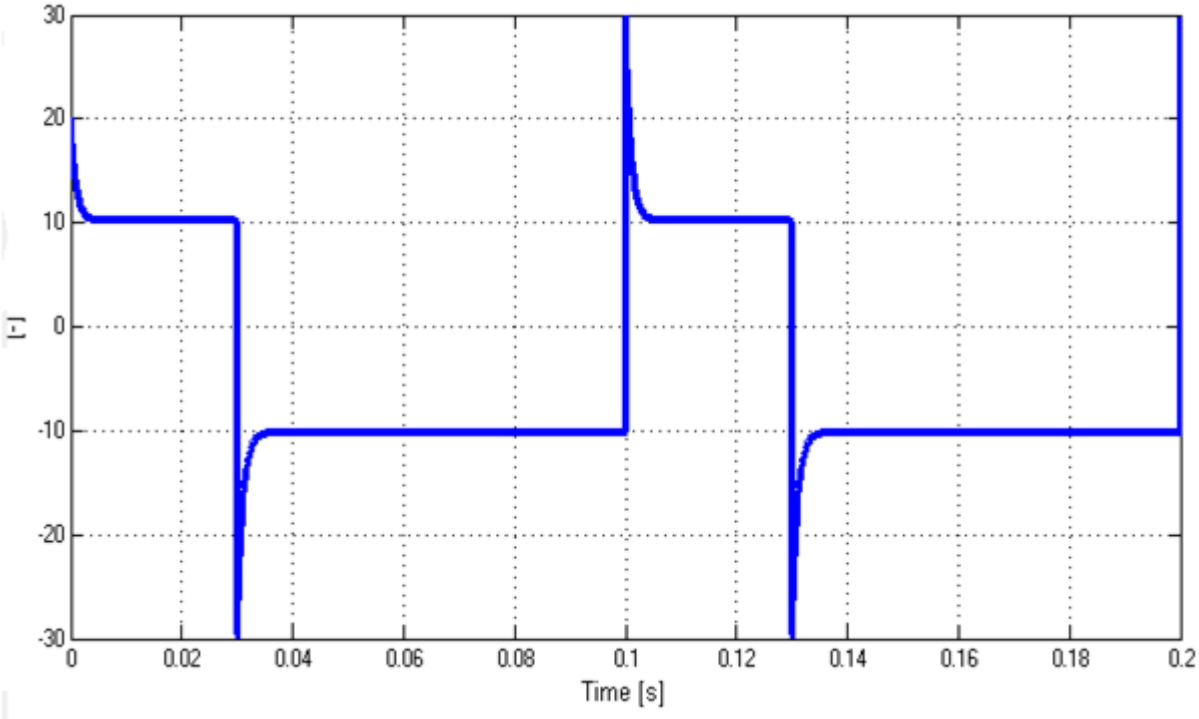
Další simulace je pro znázornění „mezi průběhů“ jak to vypadá se signálem označeným jako výstup z integrátoru a také jak vypadá signál před saturačním omezením.Proporcionální omezení je nastaveno na hodnotu 20 a integrační časová konstanta je rovna 5ms.



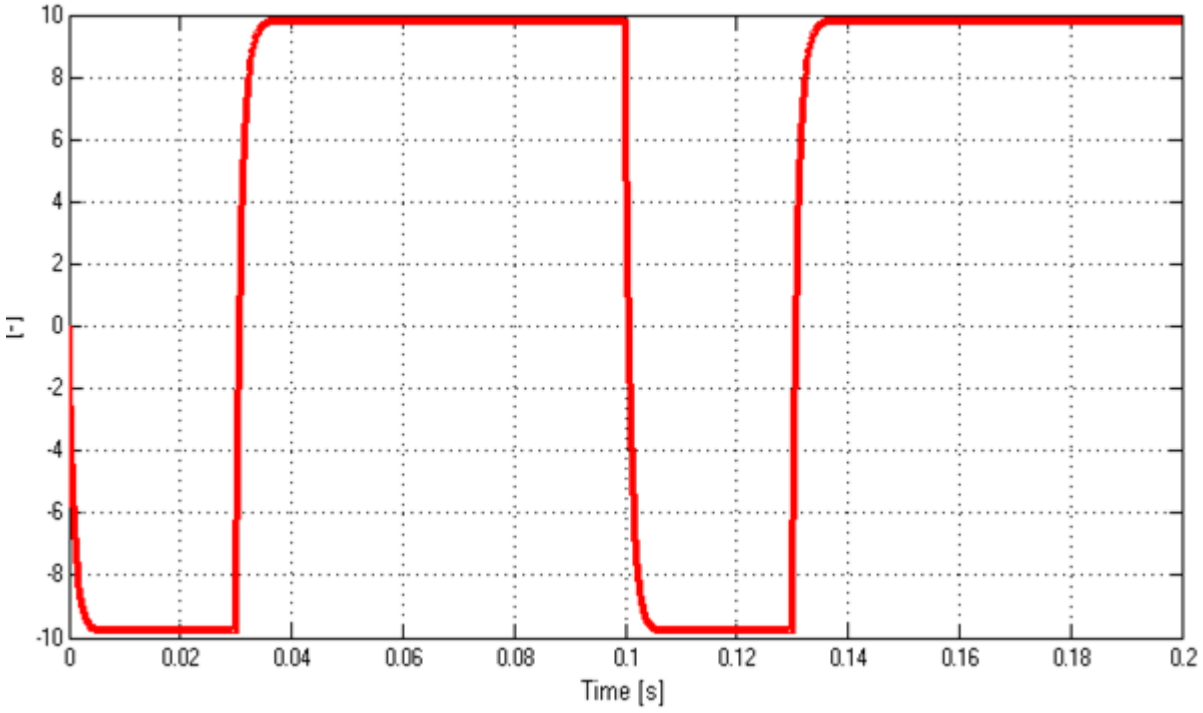
Obr.35 Simulační model PI regulátoru



Obr.36 Výstup z PI regulátoru



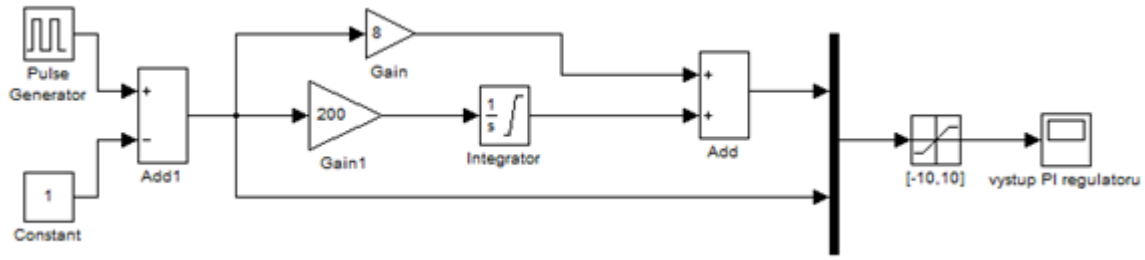
Obr.37 Výstup před omezením regulátoru



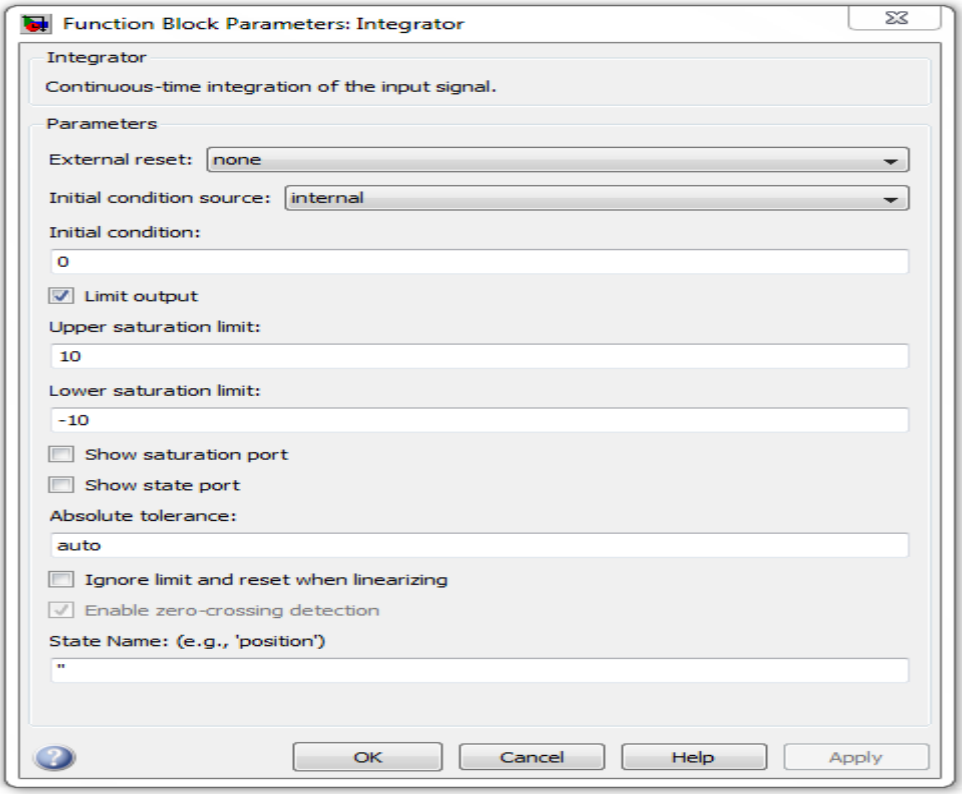
Obr.38 Výstup z integrátoru

4.3 Simulace s omezením integrační složky pomocí základního bloku Integrator

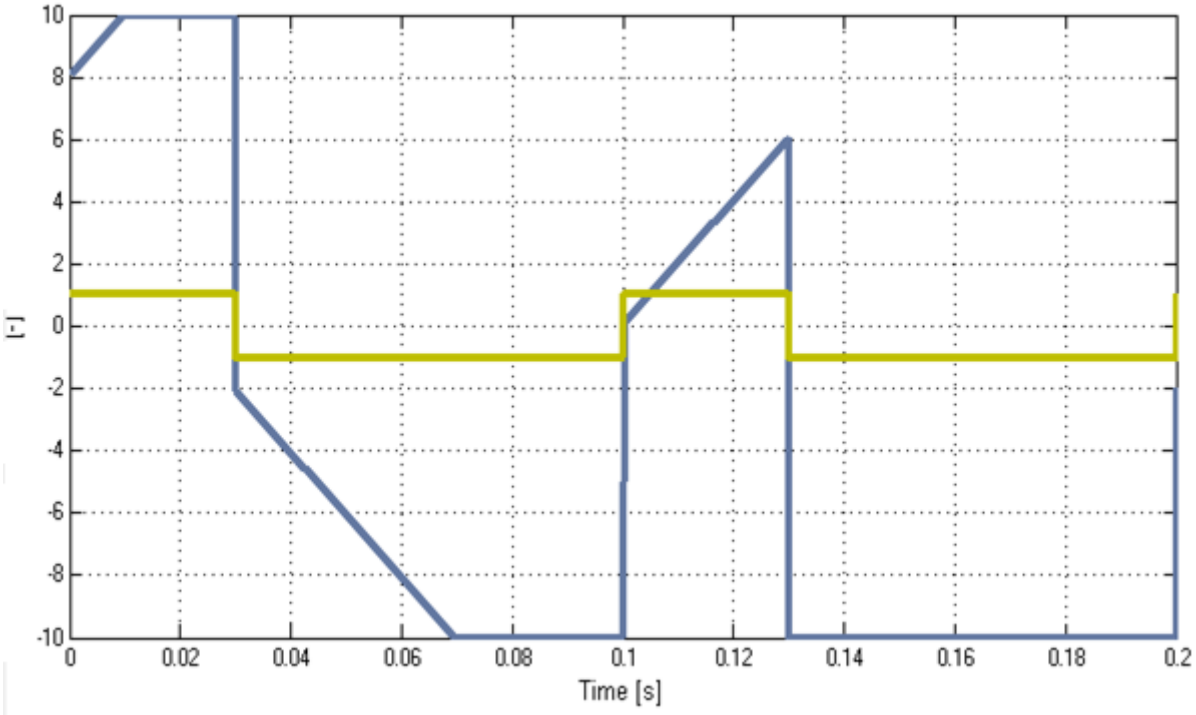
Simulace omezení integrační složky pomocí základního bloku Integrator, který je obsažen v knihovně Continuous.V této simulaci není provedena vnitřní integrace (blok Integrator neumožňuje tuto možnost).



Obr.39 Simulační model PI regulátoru

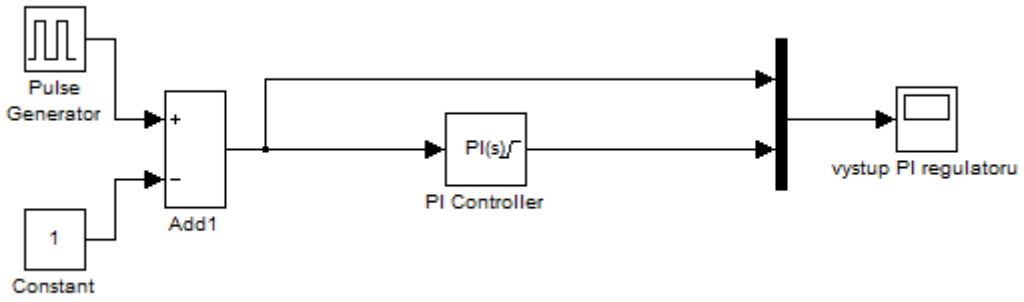


Obr.40 Hodnoty nastavení parametrů bloku Integrator

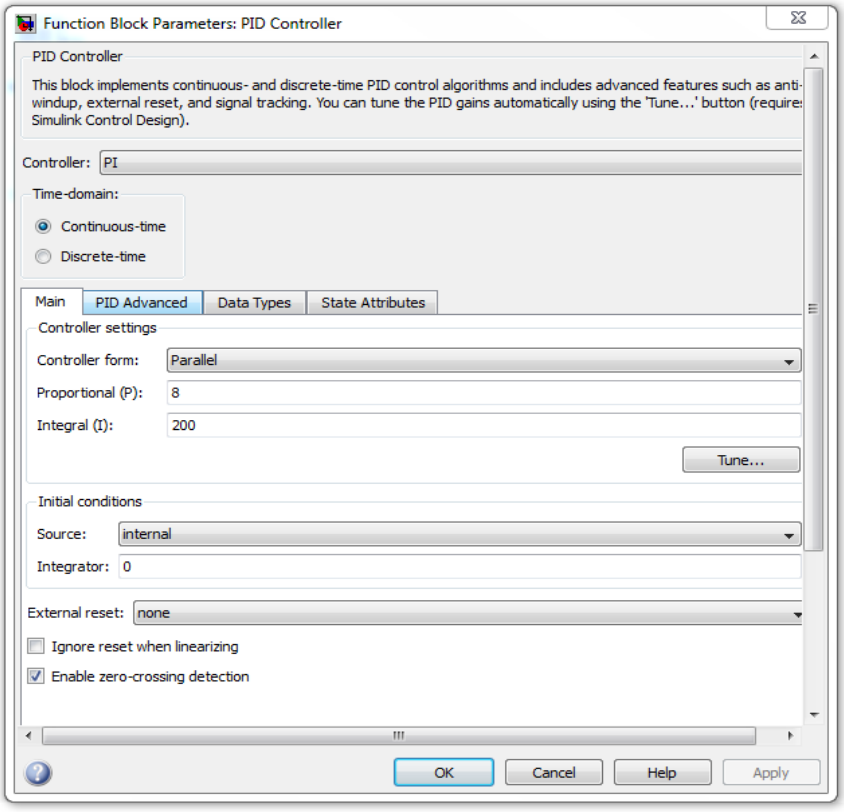


Obr.41 Výstup z PI regulátoru

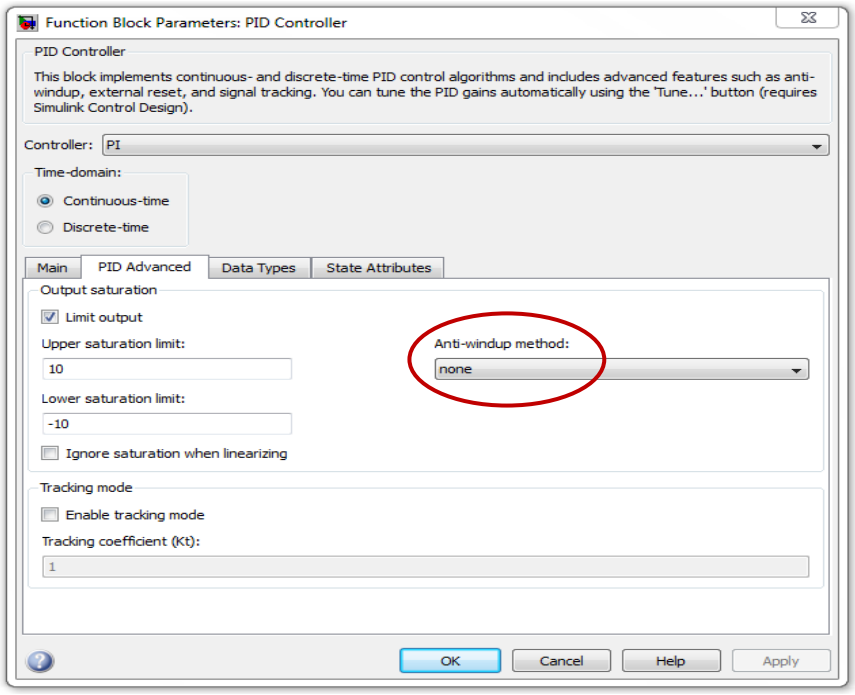
Simulace hotového bloku PI regulátoru, kde je nastavení omezení integrační složky, ale není už nastaveno omezení pomocí metody Back-calculation.



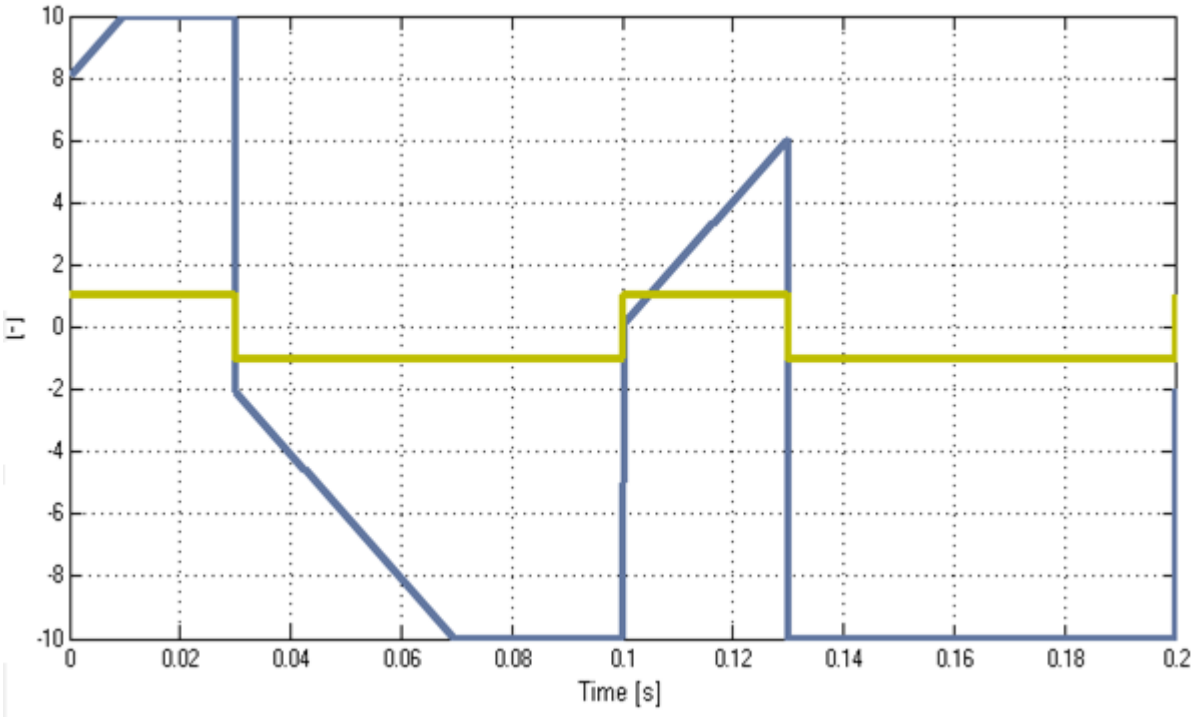
Obr.42 Simulační model PI regulátoru



Obr.43 Hodnoty nastavení vstupního PI regulátoru



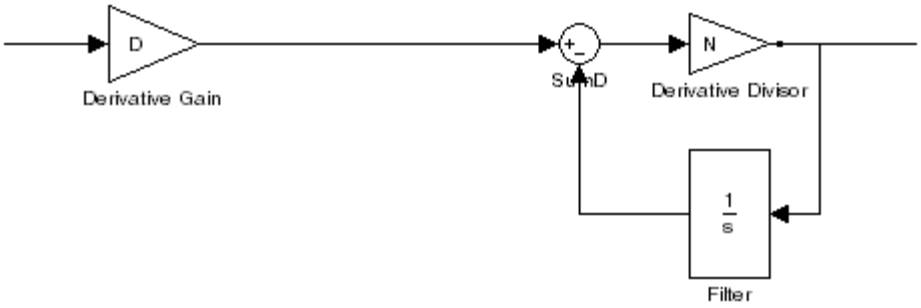
Obr.44 Hodnoty nastavení bez omezení pomocí Anti-windup metody



Obr.45 Výstup z PI regulátoru

4.4 Simulace přechodové charakteristiky v závislosti na změně filtračního koeficientu (N)

Při nastavování parametrů bloku PID regulátoru lze nastavit také parametr označovaný jako filtrační koeficient – N.Tento koeficient je dostupný pouze pro regulátory typu PID a PD, protože je vázán na derivační složku regulátoru.



Obr.46 Blokové schéma derivační složky s filtračním koeficientem

Z uvedeného blokového schématu si můžeme odvodit Laplaceův přenos a postupnou úpravou získáme filtrační konstantu, kterou si můžeme označit τ_f.

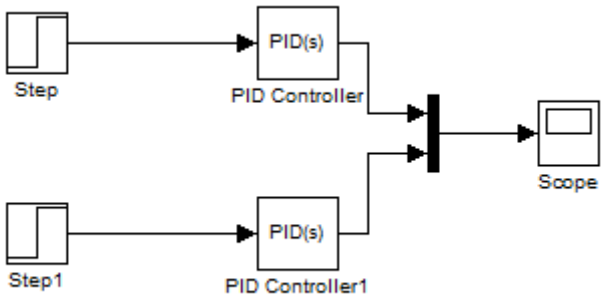
$$D \cdot \frac{N}{1+\frac{N}{s}} \qquad (4.4 - 1)$$

Po úpravě dostaneme:

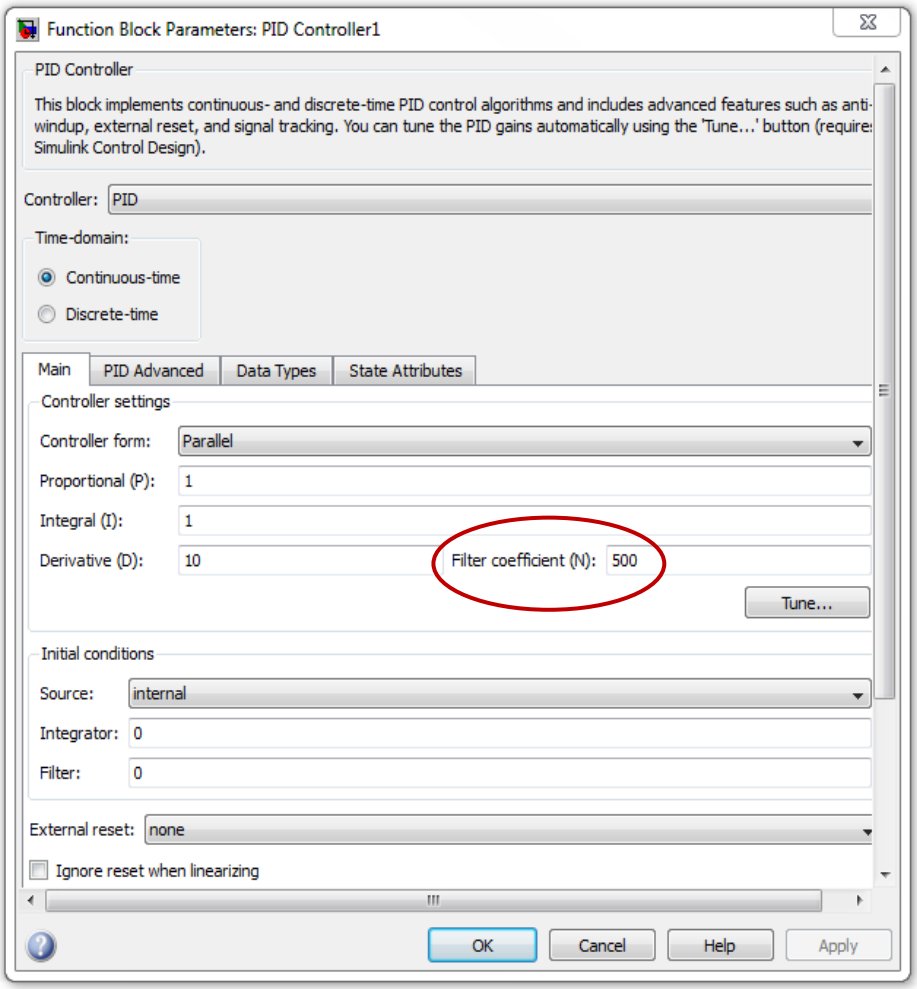
$$\frac{Ds}{1+s\cdot\frac{1}{N}} \qquad (4.4 - 2)$$

Kde 1/N je právě filtrační konstanta τ_f

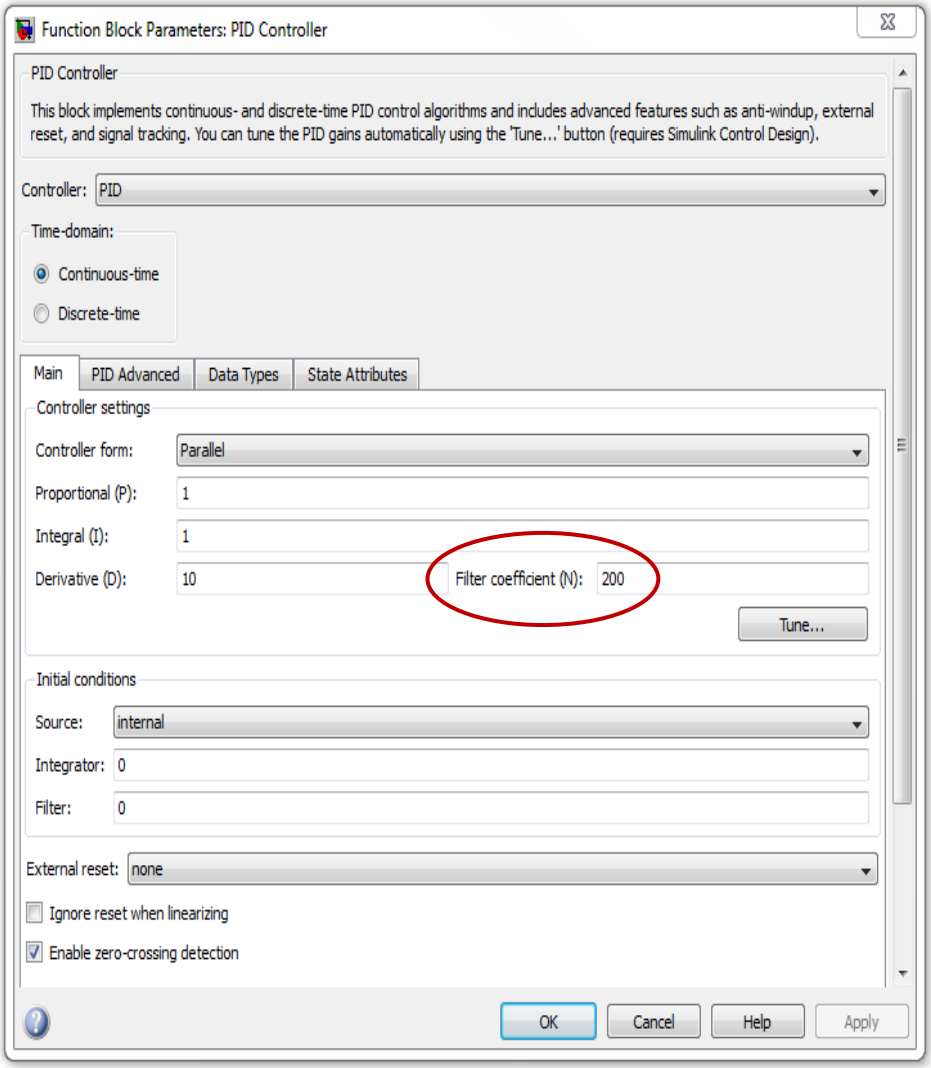
Nyní můžeme nasimulovat dva průběhy pro dvě různé hodnoty filtračního koeficientu N.



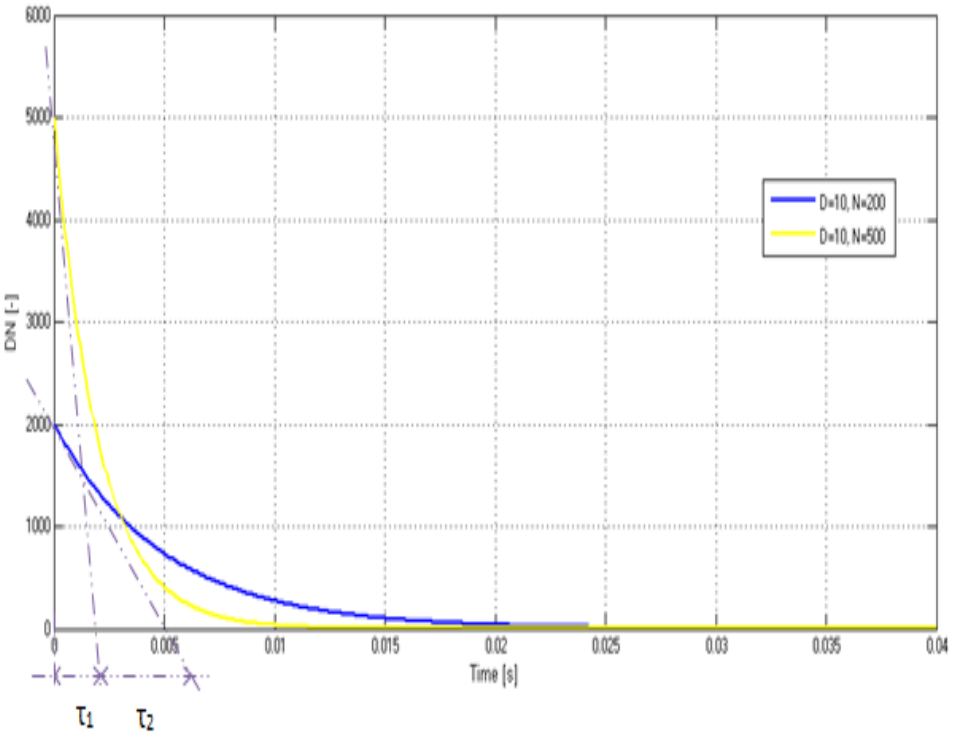
Obr.47 Blokové schéma pro určení přechodových charakteristik v závislosti na změně filtračního koeficientu N



Obr.48 Nastavení PID regulátoru pro určení závislosti filtračního koeficientu N



Obr.49 Nastavení PID regulátoru pro určení závislosti filtračního koeficientu N



Obr.50 Přechodová charakteristika pro dvě různé hodnoty filtračního koeficientu

Z přechodové charakteristiky, může vyčíst dvě hodnoty filtračních konstant τ_1 a τ_2 pro dvě různé hodnoty N.

$$\tau_1 = \frac{1}{N} = \frac{1}{500} = 0{,}002s$$

$$\tau_2 = \frac{1}{N} = \frac{1}{200} = 0{,}005s$$

Přechodová charakteristika ideálního D členu je rovna odezvě na Diracův impuls, který si můžeme představit jako nekonečně úzký a zároveň nekonečně vysoký impuls.Z tohoto si můžeme odvodit, že filtrační koeficient je velmi důležitou vlastností, která nám udává hodnotu zpoždění.

4.5 Zhodnocení a porovnání simulací

První simulace PI regulátoru se zabývá vnitřním omezením integrační složky pomocí bloku Fcn obsaženého v programu Matlab – Simulink (obr.25), kde celý model je sestaven ze základních bloků.Ve druhé simulaci je místo bloku Fcn použita simulace pomocí metody back – calculation (obr.29), což je dynamický přepočet integračního vstupu pomocí časové konstanty Tt a pro simulaci slouží rovněž model složený se základních bloků. Jak můžeme z obou výstupů vidět, tak zvolená metoda back – calculation nám v tomto případě provedla bezproblémové vnitřní omezení integrační složky, stejně jako blok Fcn.

Další simulace, která slouží ke stejnému účelu, tedy vnitřnímu omezení integrační složky je sestavena z již hotového bloku PI regulátoru (obr.31), ve kterém si můžeme zvolit meze saturace a také si zvolit jakou metodou omezení chceme provést tzv. anti – windup, tedy omezení akční veličiny na průběh regulace. Z výsledného průběhu můžeme vidět, že se nijak neliší od předchozích simulací vytvořených ze základních bloků, z čehož vyplývá, že pro simulaci je efektivnější použít již hotový blok regulátoru. Dále jsem zobrazil „mezi průběhy“ pro lepší představu jak to vypadá na výstupu integrátoru a také před samotným omezením regulátoru.

Následně jsem pro porovnání provedl simulace s omezením integrační složky, ale bez vnitřního omezení (obr.39,42). Z průběhů můžeme vidět, že regulátor je sice omezen, ale vnitřní integrace pokračuje dále, což vede ke zpomalení celého procesu a výsledkem může být nepřesně navržený regulátor, což je nežádoucí a při návrhu regulátoru to musíme brát v potaz.

Poslední simulací je simulace přechodové charakteristiky filtračního koeficientu (obr.47). Z výsledné přechodové charakteristiky lze vyčíst, že čím větší je filtrační koeficient, tím se výsledná charakteristika blíží ideálnímu derivačnímu členu, který je roven odezvě na Diracův impuls.

5. Závěr

Úkolem této bakalářské práce bylo seznámit se blíže s bloky pro řízení elektrických pohonů v programu Matlab – Simulink a následně vybrané bloky použít v simulacích a vyzkoušet chování při měnících se parametrech vybraných bloků.

V první spíše teoretické části, jsem vybral vhodné bloky pro řízení elektrických pohonů, u kterých jsem následně detailně popsal všechny možnosti nastavení, kde vodítkem mi byla nápověda každého bloku, která je v programu Matlab – Simulink uvedena. Popsané bloky jsem si pro přehlednost rozdělil podle typu systémů (spojitá a diskrétní oblast) tak jak to člení i samotný program Matlab – Simulink.

Pro druhou „simulační“ část jsem si vybral simulaci PI regulátoru, vzhledem k tomu, že patří k nejpoužívanějším typům regulátorů v technice elektrických pohonů a tento regulátor jsem zkoumal převážně z hlediska vnitřního omezení integrační složky, což lze provést pomocí standardních bloků nebo také inteligentněji pomocí bloku PI regulátoru obsaženého v Matlab - Simulinku pomocí omezení anti-windup tzv. metodou back-calculation.
V další části simulace jsem se zabýval, jak se chová PI regulátor, který má omezení integrační složky, ale není provedeno vnitřní omezení tak jako v předcházejícím případě pomocí standardních bloků a také hotového bloku regulátoru.
Poslední část simulace se zabývá simulací přechodové charakteristiky PID regulátoru v závislosti na změně filtračního koeficientu derivační složky.

Další možností rozšíření této práce se nabízí ve spojitosti s porovnáním chování regulátorů ve spojité a diskrétní oblasti. Také lze provést porovnání metody back-calculation a metody clamping, která je také obsažena v hotových blocích regulátoru. Dále se nabízí vyzkoušet chování celého PID regulátoru v závislosti na změně filtračního koeficientu z hlediska chování vzájemných vazeb mezi časovými konstantami integračního a derivačního členu.

Použitá literatura

[1] HUMUSOFT s.r.o. je distributor software MATLAB® a Simulink® pro Českou republiku <http://www.humusoft.cz/produkty/matlab/> [online]

[2] Ing. Libor Kupka, MATLAB & SIMULINK studijní materiály pro předmět Základy Kybernetiky [online]. 2008 <http://www.mti.tul.cz/files/zky/MATLABaSimulink.pdf>

[3] prof. Ing. Vilém Srovnal, Csc., Kybernetika [online]. 2008 <http://skola.spectator.cz/5_SEMESTR/Kybernetika/Kyb%E8a.pdf>

[4] prof. Ing. Brandštetter Pavel, CSc., technické prostředky pro řízení elektrických pohonů (učební texty pro kombinované a distanční studium) [online]. 2005 <http://fei1.vsb.cz/kat430/old/Studium/Materialy/TPREP/TPREP%20cast%201.pdf>

[5] The MathWorks, Inc. Matlab [počítačový program]. Version 7.9.0 (R2009b) for Windows. Natick (Massachusetts, US)

[6] Miroslav Kirchner, Porovnání diskrétního a spojitého regulátoru při přímovazební a zpětnovazební regulaci [online]. 2009 <http://dspace.k.utb.cz/bitstream/handle/10563/10048/kirchner_2009_bp.pdf?sequence=1>

[7] Bc. Radek Kundera, řízení systémů při omezení akčních veličin [online]. 2008 <http://dspace.k.utb.cz/bitstream/handle/10563/5099/kundera_2008_dp.pdf?sequence=1>

Přílohy

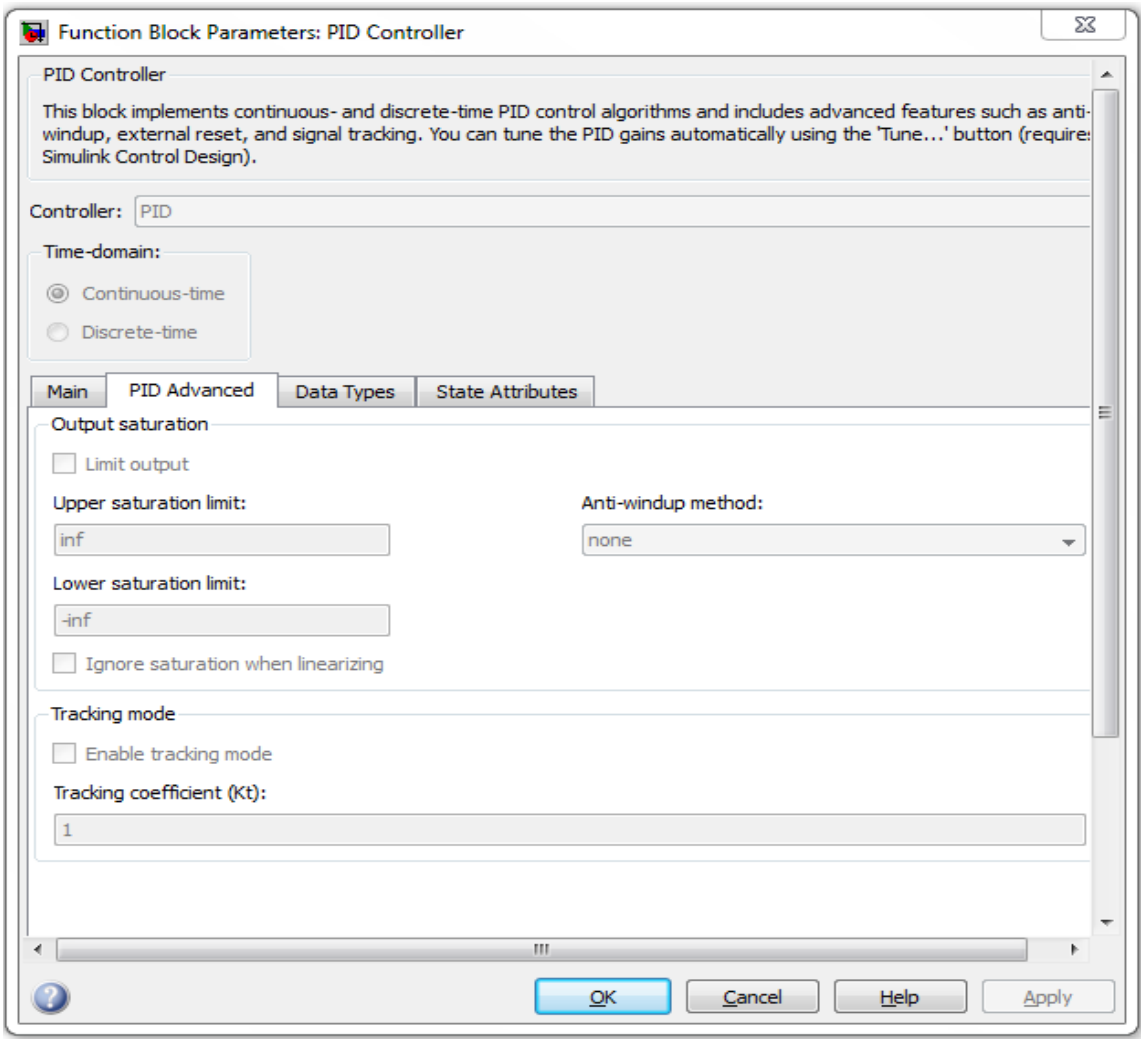
P1 – Popis zbývajících záložek PID regulátoru

(PID Advanced, Data Types, State Attributes)

P1 – Popis zbývajících záložek PID regulátoru

(PID Advanced, Data Types, State Attributes)

Další záložku, kterou popíšeme je záložka *PID* **Advanced**:



Obr.1 Okno pro nastavení parametrů bloku PID Controller – PID Advanced

Limit output (limity saturace) - defaultně je tato možnost vypnuta. Aktivací této možnosti lze nastavit vnitřní omezení (saturaci) bloku interně. Omezuje výstupní blok na dolní nebo horní mezi nasycení, vždy pokud součet překročí tyto limity.

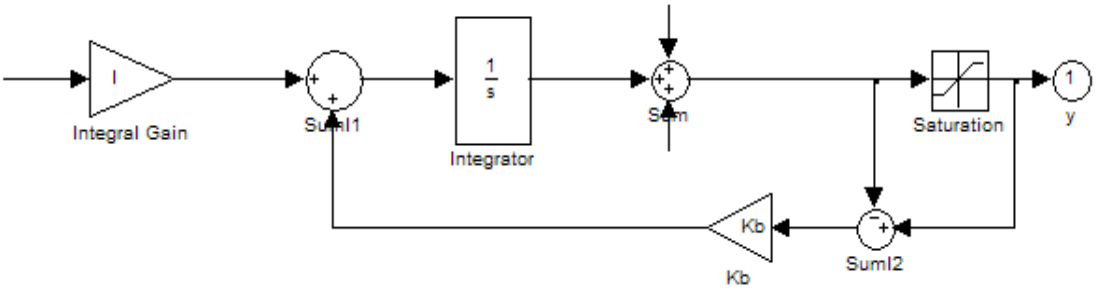
- *Lower saturation limit* – specifikuje dolní mez výstupního bloku. Výstupní blok na dolní mezi nasycení je dán součtem proporcionální, integrační a derivační složky.Defaultně je hodnota –inf.

- *upper saturation limit* - specifikuje horní mez výstupního bloku. Výstupní blok na horní mezi nasycení je dán součtem proporcionální, integrační a derivační složky. Defaultně je hodnota –inf.

- *Anti-windup method* (problém přetékání inegrátoru) – je to mechanismus, který vyprázdní integrační složku, když je blok nasycený, k tomu dochází v případě kdy součet složek v bloku přesahuje výstupní limity. Hodnoty můžeme nastavit:

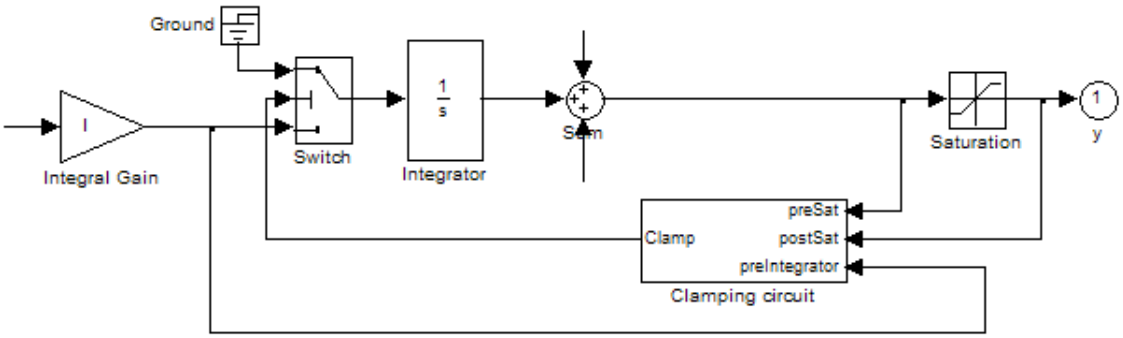
- *none* – což je defaultní stav, a v tomto nastavení není metoda využita.Toto nastavení může způsobit neohraničené vnitřní blokové signály, když výstup se zdá být ohraničen saturačními limity.Toto může mít za následky pomalé zotavení z nasycení nebo neočekávané přetékání.

- *back-calculation* (zpětný přepočet) – vyprázdní integrační složku, jakmile je výstupní blok nasycen.



Obr.2 Schématický blok metody back-calculation

- *clamping* (blokování) - zastaví integraci, když součet hodnot v bloku přesahuje výstupní limity a integrační výstup a vstupní blok mají stejné znaménko. Integrace pokračuje, když znaménko je opačné.



Obr.3 Schématický blok metody clamping

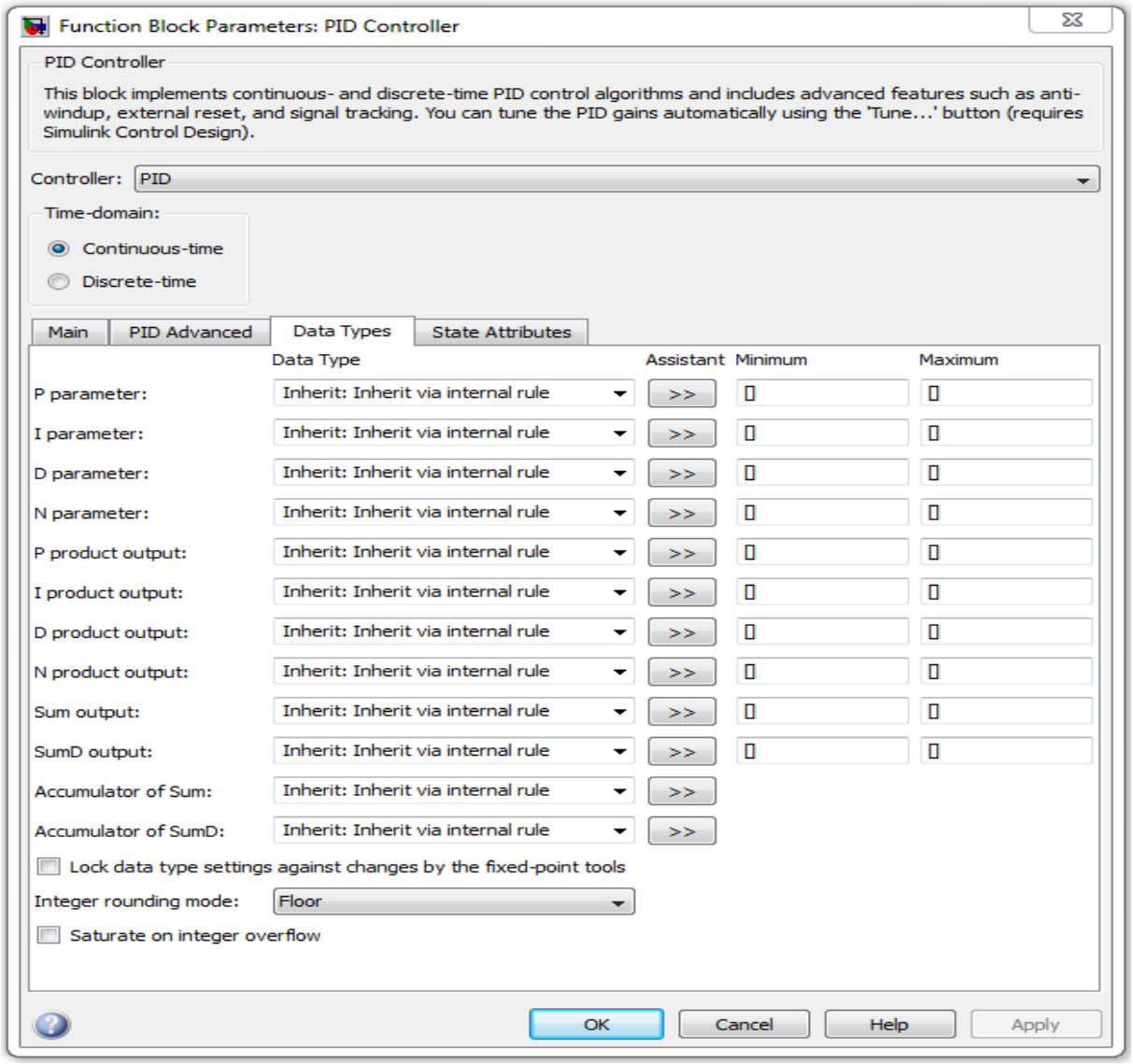
Ignore saturation when linearizing (ignorování saturace při linearizaci) – donutí pomocí linearizačních příkazů ignorovat meze výstupního PID regulátoru. Ignorováním výstupních limitů

umožňuje linearizaci modelu kolem pracovního bodu i v případě, že pracovní bod způsobuje překročení výstupních limitů PID regulátoru. Defaultně je tato možnost vypnuta.

Enable cracking mode – aktivuje sledovací signál, který umožňuje výstupu z PID regulátoru sledovat tento signál. Když je tento signál aktivní, tak rozdíl mezi sledovaným signálem a výstupním blokem je vracen zpět do integrátoru se ziskem Kt. Defaultně je tato hodnota vypnuta. Hodnota Kt je nastavena na hodnotu 1.

Tracking coefficient (Kt) – (sledovací koeficient) - Určuje Kt, což je zesílení signálu zpětné vazby sledování smyčky.

Další záložku, kterou popíšeme je záložka **Data Types**:



Obr.4 Okno pro nastavení parametrů bloku PID Controller – Data Types

Simulink podporuje všechny datové typy kromě int64 (64-bitové číslo se znaménkem) a uint64 (64-bitové číslo bez znaménka). Datové tipy můžeme nastavit následujícím parametrům: P, I, D, N dále výstupním hodnotám P, I, D, N, dále součtu výstupu a nebo saturačnímu výstupu. Seznam možných datových typů ukazuje tabulka níže.

Name	Description
double	Double-precision floating point
single	Single-precision floating point
int8	Signed 8-bit integer
uint8	Unsigned 8-bit integer
int16	Signed 16-bit integer
uint16	Unsigned 16-bit integer
int32	Signed 32-bit integer
uint32	Unsigned 32-bit integer

Obr.5 Tabulka podporovaných datových typů v programu Simulink

Defaultní datový typ je nastaven na hodnotu Inherit via internal rule (převzato z vnitřních pravidel). Simulink zvolí kombinaci výstupu datového typu, který vyžaduje nejmenší množství paměti. Ještě lze nastavit datový typ na hodnotu Same as input, což znamená, že je použit datový typ, jaký je na výstupu.

V záložce data Types, jsou ještě další možnosti.

- *Lock data type settings againts changes by the fixed-points tools* (zamyká nastavení datového typu proti změně pevně stanového nastavení) – defaultně je tato možnost vypnuta.

- *Saturate on integer overflow* (nasycení celočíselného přesycení) – defaultně je tato možnost vypnuta.

- *Integer rounding mode* (celočíselný zaokrouhlovací režim) – defaultně nastaveno Floor.

- *Ceiling* (horní mez) – zaokrouhlená kladná a záporná čísla směrem k plus nekonečnu.

- *Convergent* (sbíhavý) – zaokrouhlené číslo nejblíže reprezentované hodnotě, jestliže existuje zaokrouhlí se na nejbližší uložené hodnotě.

- *Floor* (dno) - zaokrouhlená kladná a záporná čísla směrem k zápornému nekonečnu.

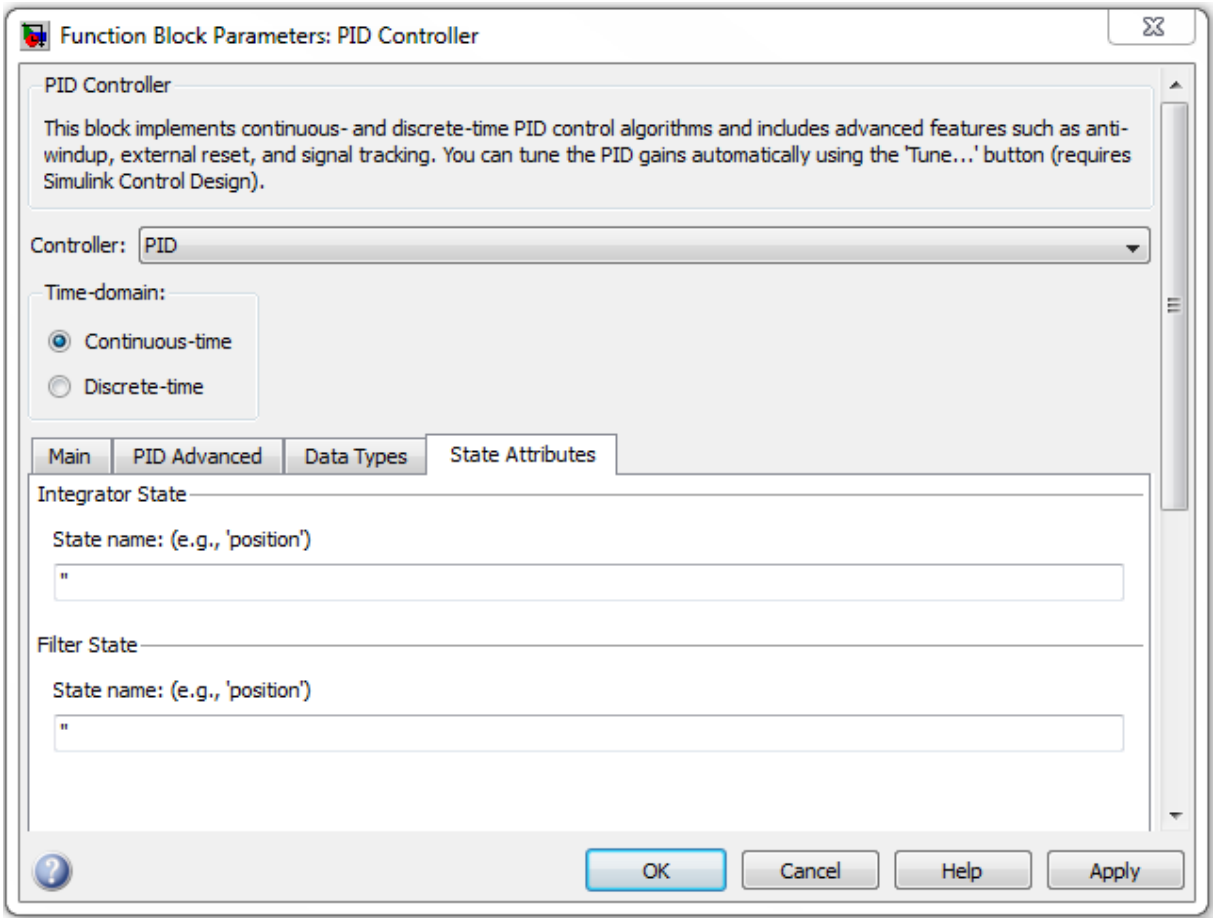
- *Nearest* (nejbližší) – číselný režim nejbližší reprezentované hodnoty.

- *Round* (zaokrouhlené číslo) – zaokrouhlené číslo nejblíže reprezentovatelné hodnotě, jestliže existuje kolem kladných čísel plus k nekonečnu.

- *Simplest* (jednoduchost) - Tato volba umožňuje optimalizaci zaokrouhlování pro více bloků.

- *Zero* (nula) – zaokrouhlování čísla kolem nuly.

Poslední nastavitelnou položkou u PID regulátoru je **State Attributes**:



Obr.6 Okno pro nastavení parametrů bloku PID Controller – State Attributes

- *State name* (uvedení názvu) – přiřazuje unikátní názvy každému stavu. Stavové jméno se vztahuje pouze k určitému bloku.Název se zapisuje do uvozovek. Defaultně není nastaven žádný název.